

مدخل إلى علم الحاسوب والبرمجة بلغة باسكال

د. غازي إبراهيم رحو
أستاذ مشارك

د. محمد نوفل ضاهر
أستاذ مساعد

د. إبراهيم عبد الواحد نائب
أستاذ مشارك

قسم الحاسوب وأنظمة المعلومات
جامعة العلوم التطبيقية



مدخل إلى علم الحاسوب
والبرمجة بلغة باسكال

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



مكتبة واداء الوثائق
الوطنية

هاتف ٤٦٥٠٦٢٤ - فاكس ٤٦٥٠٦٢٤ (٩٦٢ ٦)

ص.ب ٢١٥٣٠٨ عمان ١١١٢٢ الأردن

حقوق الطبع محفوظة

الطبعة الاولى

١٤٢٠ هـ - ١٩٩٩ م

رقم الايداع لدى دائرة المكتبات والوثائق الوطنية

(١١٩٠ / ٧ / ١٩٩٩)

رقم الإجازة المتسلسل لدى دائرة المطبوعات والنشر

(٧٨٦ / ٧ / ١٩٩٩)

الإهداء

إلى من صبروا ... ووقفوا إلى جانبنا طيلة

مراحل اعداد هذا الكتاب

زوجاتنا ... أطفالنا

مع خالص الحب والتقدير عرفانا منا لهم

المؤلفون

شكر

يتقدم المؤلفون بجزيل الشكر والعرفان لكل من تقدم باقتراحات وإرشادات وملاحظات حول طباعة وإخراج هذا الكتاب ، وخاصة الأساتذة الأفاضل في قسم الحاسوب وأنظمة المعلومات - جامعة العلوم التطبيقية .

لهم منا جزيل الشكر على ملاحظاتهم القيمة، جزاهم الله خيرا

المؤلفون

المحتويات

مقدمة

15

الفصل الأول

الحاسوب (أهميته - نشأته - تطوره)

19

(1-1) أهمية دراسة الحاسوب

19

(1-1-1) الحواسيب وأثرها على التربية والتعليم والمعالج

21

(2-1-1) الحواسيب والاتصال مع الغير

22

(2-1) نبذة تعريفية عن الحواسيب

22

(1-2-1) المفاهيم الأساسية للحاسوب

26

(2-2-1) تعريف الحاسوب وخصائصه

29

(3-2-1) أنواع الحواسيب

32

(4-2-1) استخدامات الحواسيب

34

(5-2-1) لمحة تاريخية عن تطور الحواسيب

40

(6-2-1) الأجيال المختلفة للحواسيب

47

تمارين وأسئلة عامة

الفصل الثاني

المكونات المادية للحاسوب

49

(1-2) مقدمة

51

(2-2) وحدة المعالجة المركزية

52

(1-2-2) وحدة الحساب و المنطق

54

(2-2-2) وحدة التحكم

59

(3-2) وحدات التخزين الداخلية

59

(1-3-2) الذاكرة الرئيسية

64

(2-3-2) الذاكرة الفورية

64

(3-3-2) الذاكرة الدائمة

66

(4-3-2) المسجلات

67

(5-3-2) المخازن الانتقالية

69

(4-2) وحدات التخزين الثانوية

76	(5-2) وحدات الإدخال
81	(6-2) وحدات الإخراج
88	تمارين وأسئلة عامة

الفصل الثالث الانظمة العددية

89	(1-3) ماهية النظم العددية
89	(1-1-3) مفاهيم النظم العددية
91	(2-1-3) النظام العشري
93	(3-1-3) النظام الثنائي والعمليات الحسابية عليه
93	(1-3-1-3) مقدمة
93	(2-3-1-3) التحويل من النظام الثنائي الى العشري
95	(3-3-1-3) التحويل من النظام العشري الى الثنائي
97	(4-3-1-3) العمليات على النظام الثنائي
103	(4-1-3) النظام العد الثماني
103	(1-4-1-3) مقدمة
103	(2-4-1-3) التحويل من النظام الثماني الى النظام العشري
104	(3-4-1-3) التحويل من النظام الثماني الى النظام العشري وبالعكس
106	(4-4-1-3) العمليات في النظام الثماني
109	(5-1-3) النظام السادس عشر
109	(1-5-1-3) مقدمة
110	(2-5-1-3) التحويل من النظام السادس عشر الى العشري وبالعكس
112	(3-5-1-3) التحويل من النظام السادس عشر الى النظام الثنائي وبالعكس
113	(4-5-1-3) العمليات الحسابية في النظام السادس عشر
116	(2-3) المتممات العددية
116	(1-2-3) مفهوم المتمم العددي
116	(2-2-3) المتممات في النظام العشري
118	(3-2-3) المتممات في النظام الثنائي
121	تمارين وأسئلة عامة

الفصل الرابع

تمثيل البيانات والتعليمات داخل الحاسوب

- 123 (1-4) مفاهيم اساسية
- 123 (1-1-4) التمثيل الداخلي
- 123 (2-1-4) العنوان
- 124 (2-4) تمثيل الأعداد الصحيحة
- 125 (1-2-4) تمثيل العدد وإشارته
- 126 (2-2-4) التمثيل بالتميم الثنائي
- 127 (3-2-4) حدود التمثيل ومشكلة الفائض
- 128 (3-4) تمثيل الأعداد ذات النقطة العائمة
- 129 (1-3-4) الشكل المعياري
- 130 (2-3-4) المسجل ذو النقطة العائمة
- 132 (4-4) تمثيل الأحرف و الرموز
- 132 BCD نظام (1-4-4)
- 133 EBCDIC نظام (2-4-4)
- 133 ASCII نظام (3-4-4)
- 135 (5-4) تمثيل الأوامر والتعليمات
- 139 تمارين وأسئلة عامة

الفصل الخامس

الكيان المنطقي للحاسوب (البرمجيات)

- 141 (1-5) ماهية البرمجيات
- 141 (2-5) أنواع برمجيات الحاسوب
- 141 (1-2-5) برمجيات النظام
- 143 (2-2-5) برمجيات التطبيقات
- 145 (3-5) نظم التشغيل
- 145 (1-3-5) تعريف نظام التشغيل
- 146 (2-3-5) الخصائص العامة لنظام التشغيل
- 147 (3-3-5) أنواع نظم التشغيل

149	(4-3-5) أنواع نظم التشغيل المستخدمة مع الحواسيب الشخصية
151	(4-5) نظام التشغيل ام اس -دوس MS-DOS
152	(1-4-5) تشغيل الحاسوب
154	(2-4-5) أوامر نظام التشغيل MS-DOS
155	(1-2-4-5) أوامر الخدمات
180	(5-5) نظام التشغيل ويندوز
180	(1-5-5) تعريف وميزات النظام ويندوز
184	(2-5-5) المبادئ الاساسية لتعامل مع النظام ويندوز 95
195	(3-5-5) التعامل مع المجلدات والملفات
202	(6-5) البرمجيات التطبيقية الجاهزة
203	(1-6-5) برامج معالجة النصوص
206	(2-6-5) الجداول الإلكترونية
207	(3-6-5) برامج إدارة قواعد البيانات
211	(4-6-5) برامج الإحصاء المحوسب SPSS

الفصل السادس

الشبكات الحاسوبية وتراسل المعلومات

213	(1-6) أهمية تراسل المعلومات
214	(2-6) الاتصالات والمعلوماتية
215	(3-6) الشبكة : تعريفها-أنواعها-أهميتها
216	(4-6) مكونات أنظمة الشبكات
218	(1-4-6) معدات الشبكات
219	(2-4-6) برمجيات الشبكات
219	(3-4-6) قنوات الاتصال
222	(5-6) شبكات المعالجة الموزعة
222	(6-6) هيكلية الشبكات
223	(1-6-6) الشبكة النجمة
224	(2-6-6) شبكة الحلقة
225	(3-6-6) الشبكة الخطية

206	(4-6-6) الشبكة الهرمية
227	(7-6) الإنترنت
227	(1-7-6) ماهية الإنترنت
229	(2-7-6) لمحة تاريخية عن شبكة الانترنت
232	(3-7-6) خدمات الإنترنت
234	(4-7-6) الاتصالات والعنونة على الإنترنت
236	(5-7-6) أسس الدخول إلى شبكة الإنترنت
237	(6-7-6) البرامج المستخدمة في البحث وتصفح المعلومات عبر الإنترنت
243	تمارين وأسئلة عامة

الفصل السابع

لغات البرمجة

245	(1-7) المكونات والعناصر الأساسية للغة البرمجة
247	(2-7) أنواع لغات البرمجة
247	(1-2-7) لغات المستوى الواطئ
248	(2-2-7) لغات المستوى العالي
258	تمارين وأسئلة عامة

الفصل الثامن

برمجة الحاسوب والمخططات الانسيابية

259	(1-8) المراحل الأساسية لحل أي مسألة حاسوبياً
259	(1-1-8) المرحلة الاولى (دور الانسان)
264	(2-1-8) المرحلة الثانية (دور الحاسوب)
266	(3-1-8) المرحلة الثالثة (التوثيق)
267	(2-8) مخططات سير العمليات
267	(1-2-8) مفهوم وأهمية مخططات سير العمليات
268	(2-2-8) الأشكال والرموز الاصطلاحية المستخدمة في المخططات
269	(3-2-8) أنواع المخططات الانسيابية
270	(3-8) أنواع المخططات الانسيابية لسير العمليات في البرامج الحاسوبية
270	(1-3-8) مخططات التابع البسيط

272	(8-3-2) مخططات التفرع (الاختبار)
280	(8-3-3) مخططات التكرار
290	تمارين وأسئلة عامة

الفصل التاسع البرمجة بلغة باسكال

(أساسيات ومكونات - جمل الإدخال والإخراج - جمل التحكم والتكرار)

291	(9-1) مقدمة
293	(9-2) أساسيات ومكونات لغة باسكال
293	(9-2-1) مجموعة رموز باسكال
294	(9-2-2) الكلمات المحجوزة
294	(9-2-3) مسميات المبرمج التعريفية
295	(9-2-4) الأسماء التعريفية القياسية
296	(9-2-5) الأعداد
299	(9-2-6) الثوابت الرمزية
300	(9-2-7) أنواع المعطيات
303	(9-2-8) التعابير الحسابية والمنطقية
309	(9-2-9) قاعدة الأولوية
310	(9-2-10) جمل الإسناد
311	(9-3) المكونات الهيكلية لبرنامج باسكال
315	(9-4) أساليب إدخال وإخراج البيانات في برنامج باسكال
316	(9-4-1) جمل الإدخال
320	(9-4-2) جمل الإخراج
328	(9-5) كتابة وتصميم برنامج باسكال
332	(9-6) جمل التحكم (السيطرة)
333	(9-6-1) الجمل المركبة
333	(9-6-2) جملة التحكم الشرطية IF-THEN
335	(9-6-3) جملة التحكم الشرطية IF-THEN-ELSE
337	(9-6-4) جملة التحكم الشرطية المتداخلة

344	CASE جملة التحكم الشرطية (5-6-9)
348	GO TO جملة الانتقال (6-6-9)
348	(7-9) جمل التكرار المشروط والتكرار باستخدام العدادات
349	WHILE-DO جملة التكرار المشروط (1-7-9)
355	REPEAT UNTIL جملة التكرار المشروط (2-7-9)
361	FOR جملة التكرار (3-7-9)
365	(4-7-9) جملة التكرار المتداخلة
371	تمارين وأسئلة عامة

الفصل العاشر البرمجة بلغة باسكال

	(بيانات المستخدم التعريفية-المصفوفات-البرامج الفرعية-السجلات والملفات)
377	(1-10) بناء أنواع بيانات جديدة في لغة باسكال
377	(1-1-10) البيانات معدودة القيم
379	(2-1-10) البيانات الجزئية
381	(2-10) المصفوفات
381	(1-2-10) مفهوم المصفوفات في لغات البرمجة
382	(2-2-10) تمثيل عناصر المصفوفات في لغة باسكال
383	(3-2-10) الإعلان عن المصفوفات
384	(4-2-10) أمثلة على استخدام ومعالجة المصفوفات
388	(3-10) البرامج الفرعية
389	(1-3-10) الإجراءات
392	(2-3-10) الاقتترانات
397	(4-10) السجلات
400	(5-10) الملفات
405	تمارين وأسئلة عامة

الفصل الحادي عشر

أساسيات بيئة باسكال

- 407 (1-11) الدخول الى بيئة باسكال والخروج منها
- 408 (2-11) أقسام النافذة الرئيسية لبيئة باسكال
- 412 (3-11) تشغيل شريط القوائم
- 414 (4-11) صناديق الحوار
- 415 (5-11) بعض المهمات التي يمكن القيام بها
- 415 (1-5-11) فتح ملف جديد
- 415 (2-5-11) فتح ملف مخزن مسبقا
- 416 (3-5-11) تخزين ملف باسم جديد
- 417 (4-5-11) تخزين ملف باسم قديم
- 417 (5-5-11) طباعة ملف
- 418 (6-5-11) استخدام ميزة النسخ والقص واللصق والمسح عند كتابة البرنامج
- 419 (7-5-11) تصحيح الملف قواعديا وترجمته الى لغة الآلة
- 420 (8-5-11) تشغيل البرنامج
- 420 (9-5-11) عرض نتائج تشغيل البرنامج
- 420 (10-5-11) الخروج من بيئة باسكال
- 421 المراجع العربية
- 423 المراجع الأجنبية
- 427 الاختبارات

مُقَدِّمَةٌ

نعيش اليوم عصرًا يسمى بعصر المعلوماتية، ينصب فيه الاهتمام على جمع كم من المعلومات لتخزينها واسترجاعها وقت الحاجة بهدف الاستفادة منها لاستخلاص النتائج المفيدة في عمليات اتخاذ القرارات الرشيدة والمدرسة في شتى مجالات الحياة.

لقد فرض الحاسوب نفسه كأداة هذا العصر طوعاً أو كرهاً في شتى ميادين المعرفة، أداة لا غنى عنها للباحثين، وطلاب المعرفة، للسياسيين، والاقتصاديين، والإداريين، وللأطباء، والمهندسين ... في المؤسسة، والمصنع، والبيت، والمكتب، والعيادة ...

لقد اتجه العلماء في مناطق عديدة من العالم إلى تطوير الحاسوب وتسهيل التعامل معه و استخدامه، حيث بدأ حياته في النصف الثاني من هذا القرن ضخماً ثقيلاً لا يعمل إلا ضمن ظروف معينة، والتعامل معه كان صعباً. و يحتاج إلى تدريب وخبرة طويلة.

لم يمض على عمره أكثر من خمسين عاماً وإذا به يغدو خفيفاً صغيراً سهل التعامل معه ويمتاز جميع الناس على مختلف مستوياتهم الاجتماعية والعلمية، وحتى الأطفال.

لقد كان الهدف من اختراعه آنذاك هو القيام بأعمال حسابية بسيطة،
ليعطيك ما تريده منها في زمن قصير، يعالج البيانات مهما كان حجمها
وبسرعة فائقة، يساعد في التصميم الهندسي وفي تشخيص الكثير من
الأمراض، كما أنه يمد يد العون عند اتخاذ القرار، وأصبح أيضاً قادراً أن يحل
محل العامل في المصنع و المتجر، ناهيك عن دوره في غزو الفضاء وتطوير
الأسلحة والتواصل العلمي بين الباحثين ... وتراسل المعلومات عبر شبكات،
تدعى الشبكات الحاسوبية، تتصل مع شتى بقاع العالم.

لذا رأينا من واجبنا تعريف الطالب والقارئ بأهمية الحاسوب وإعطاءه
الأسس والمبادئ والخصائص التي يجب أن يلم بها بغية مواكبة عصر المعلوماتية
وتمكنه من تسخير إمكانيات الحاسوب الهائلة في خدمته أيأ كان موقعه. وقد
راعينا عند تصنيفنا للموضوعات التدرج العلمي في الشرح والعرض المبسط،
بحيث يحقق للقارئ ما يجب أن يلم به للاستفادة من معلومات هذا الكتاب.

يتضمن هذا الكتاب أحد عشر فصلاً، نقدم في الفصل الأول شرحاً
عن أهمية الحاسوب و تطوره وأنواعه، وفي الفصل الثاني نتعرض إلى مكونات
الحاسوب المادية (الذاكرة، و المعالج، و وحدات التخزين الرئيسية، والثانوية،
وحدات الإدخال والإخراج)، أما الفصل الثالث فيتضمن الأنظمة العددية التي
يعتمد عليها الحاسوب في معالجة البيانات ثم يبين في الفصل الرابع كيفية تمثيل
البيانات حاسوبياً من أعداد ورموز وتعليمات أما في الفصل الخامس فقد
تناولنا موضوع الكيان المنطقي للحاسوب وبرمجياته المختلفة وأنظمة التشغيل،
وفي الفصل السادس أعطينا فكرة عن الشبكات الحاسوبية وشبكات الأنترنت
وتراسل المعلومات، وفي الفصل السابع تناولنا لغات البرمجة وأنواعها، أما في

الفصل الثامن فقد حرصنا على تعليم الطالب و القارئ كيفية معالجة المسائل والمشاكل حاسوبياً من خلال مراحل متسلسلة مروراً بالتحليل المنطقي للمشكلة ثم رسم مخطط سير العمليات، ثم استخدام إحدى لغات البرمجة في التفاهم مع الحاسوب بغية معالجة البيانات والحصول على المعلومات المفيدة. وقد كان التركيز في الفصلين التاسع والعاشر على لغة باسكال (PASCAL) كإحدى اللغات التي أثبتت كفاءتها وسهولة استخدامها واعتمادها على مفهوم البرمجة المهيكلية، كما ركز الفصل الحادي عشر على كيفية التعامل مع شاشة باسكال وادخال البرامج وتنقيحها.

لقد وضعنا في نهاية كل فصل عدداً كافياً من المسائل والأسئلة غير المحلولة المتدرجة في صعوبتها رغبةً منا في إتاحة المجال للطالب والقارئ لاختبار مدى فهمه لمحتويات هذا الكتاب.

وعلى الرغم من أن هذا الكتاب عمل جماعي ومسؤولية مشتركة، إلا أن أمانة التأليف تقتضي تخصيص دور كل مؤلف على حدة، فقد قام الدكتور ابراهيم نائب بكتابة الفصول (الأول، الخامس، السادس، السابع، الثامن، الحادي عشر)، وقام الدكتور غازي رحو بكتابة الفصول (التاسع، العاشر)، في حين قام الدكتور محمد ضاهر بكتابة الفصول (الثاني، الثالث، الرابع، السادس).

وإننا إذ نضع هذا الكتاب بين أيدي طلابنا وزملائنا الأعزاء، لا ندعي أننا وصلنا به إلى درجة الكمال، ولكننا نأمل أن تكون المواضيع التي يحتويها هذا الكتاب وطريقة عرضها مفيدة ومفهومة للطالب وجميع المهتمين بالدراسات المعمقة في المواضيع المتقدمة في علم الحاسوب والتي لم يتسع المجال

للخوض في تفاصيلها، وأن نكون قد وفقنا من خلال هذا الجهد المتواضع
بوضع لجنة تأسيسية في بناء مكتبة عربية تزخر بالمؤلفات العلمية في هذا
المضمار.

نود هنا أن نتوجه بالشكر إلى جميع من ساهم بشكل مباشر وغير
مباشر في إخراج هذا الكتاب، ونرجو من زملائنا الكرام إغناء تجربتنا هذه
بملاحظاتهم كي نأخذ بها عند إعادة طباعته، إن شاء الله.
والله ولي التوفيق

المؤلفون

الفصل الأول

الحاسوب (أهميته - نشأته - تطوره)

(1 - 1) أهمية دراسة الحواسيب

إن أهمية دراسة ومعرفة الحواسيب وأنواعها ومكوناتها من معدات وأجهزة وبرمجيات ولغات مختلفة يعطي فوائد كثيرة يستطيع انسانا العربي أن يجنيها من تعلم ودراسة هذه المادة التي دخلت في جميع مناحي الحياة ، حيث تساعد هذه المعرفة الإنسان على تكوين مستوى تعليمي رفيع من خلال تعلم الحواسيب وعملها واستخداماتها وتساعد في الحصول على إنتاجية عالية وإنجاز دقيق للأعمال وبكفاءة مرتفعة وقدرة على إتمام الكثير من الوظائف والتي قد لا يمكن للإنسان أن ينجزها باستخدام الأيدي أو العقل ، بالإضافة إلى تطوير النظم المختلفة وتجميع قطاعات الحياة ومعالجة البيانات وإدارة هذه البيانات بطرق سهلة وسريعة وتكلفة قليلة ، بالإضافة إلى تحسين النوعية وخاصة في الإنتاج وإدارة العمليات والمعلومات ومعالجة البيانات وكذلك ما يتعلق بالاتصالات التي أصبحت أسلوب مهم وفعال في نقل المعلومات بالإضافة إلى ما يتعلق بالتعليم وتطوره وزيادة معدلات التطور ومدى تأثيره على حياة البشرية وفي جميع القطاعات ، ولهذا فإن أهمية دراسة الحواسيب لها تأثير مباشر على حياة الإنسان وتطوره وتقدمه .

(1 - 1 - 1) الحواسيب وأثرها على التربية والتعليم والمعالجة

إن التربية والتعليم هما عاملان أساسيان في تقدم الأمم وإن الشعب المتعلم يخلق حضارة جديدة ويبني الأمم ولهذا فإن الحصول على مستوى تعليمي مرتفع

يمكن أن يأتي من خلال تعلم الحواسيب التي تؤثر على حياة البشر وذلك كونها مستخدمة في جميع قطاعات الحياة ابتداءً من البنوك والمستشفيات والمصانع والطائرات والمؤسسات المدنية والعسكرية وصولاً إلى تحديد الأمراض وتحديد مستقبل الحياة ، لذلك فإن تعلم الحواسيب ومعرفتها بصورة جلية والدخول في معرفة العديد من مظاهر التكنولوجيا الموجودة في جميع مناحي الحياة ، ومع تزايد تطور الحياة زادت الحاجة للمعرفة والتعلم بماهية هذا الجهاز وكيفية استخدامه ، ولهذا نرى أن الأمم اتجهت في إدخال الحواسيب في التعليم و التعلم لأبنائها ومنذ المراحل الدراسية الأولى وصولاً إلى شعب متعلم يستطيع معالجة جميع مشاكله من خلال استخدام الحواسيب، إذ أن الحواسيب لا تقوم بتنفيذ بعض المهام اليومية بشكل أفضل وبكفاءة أعلى فحسب ، وإنما يمكن للحواسيب أن تقوم بتنفيذ المهام التي لا يستطيع الإنسان الاعتيادي أن يقوم بها فأنظمة المحاكاة والتنبؤ في الحواسيب تستطيع أن تقدم للإنسان ما لا يستطيع الوصول إليه الإنسان نفسه بالسرعة والجهد المبذول ، فالحواسيب يمكن أن تجري عمليات كيميائية معقدة ، والوصول إلى نتائج عالية المستوى دون الخوض في التطبيق من خلال القيام بمزج أنواع مختلفة من العناصر الكيميائية ، وإنما من خلال قيام نظام أو برنامج محدد يحاكي عمليات المرج الفعلية ويعطي نتائج دقيقة وعالية المستوى ، ويستطيع الحاسوب عرض النتائج وتسجيلها وتسلسلها ، ولهذا يستطيع الحاسوب أن يقوم بالتجارب ، ويعطي النتائج الدقيقة ، كما يمكن للحاسوب أن يقوم بمهام عملية التنبؤ للمستقبل ويقوم بعمليات التحليل وإعطاء النتائج وبهذا يستطيع مساعدة الطالب في التعلم والتدريب دون المخاطرة بالمواد والمعدات ، وكذلك فإن تعلم الحواسيب يساعده في تنفيذ وظائف معينة بكفاءة عالية فاستخدام معالج النصوص يعمل على إعداد الوثائق ويساعد في تنظيمها وتسهيلها وسرعتها ودقتها ، وكما وأن المرونة في استخدام الحاسوب تكون دائماً أعلى وأكبر من استخدام الآلة الكاتبة مثلاً ،

بالإضافة إلى إمكانية استخدام القاموس الداخلي Spelling Checker الذي يساعد كثيراً في تقليل الأخطاء بالإضافة إلى البحث عن الكلمات والذي يساعد في جعل النص المستخدم ذا قوة عالية في المعنى ، هذا بالإضافة إلى أن تعلم استخدام الحواسيب تعمل على توفير الوقت والجهد والسرعة .

(1 - 1 - 2) الحواسيب والاتصال مع الغير

Computer and Communication

بعد انتشار استخدام الحواسيب في مختلف المجالات لغرض زيادة وتحسين الأداء في العمل وزيادة كفاءة الأعمال وسرعتها ، وبطور ذلك الانتشار وبزيادة ذلك الاستخدام ازدادت الحاجة إلى تبادل المعلومات والبيانات بين الوحدات المكونة لأي مؤسسة أو بين مجموعة مؤسسات ، وحيث كان نقل البيانات في السابق وتناقلها والاتصال مع الغير يتم من خلال أسلوب قديم عن طريق نقل البطاقات المثقبة أو الأشرطة المغنطة والتي كان يتم نقلها بالوسائل التقليدية ولكن مع مرور الوقت تطورت طريقة نقل البيانات وذلك بتطور الاتصالات سواء كانت هذه الاتصالات شبكات هاتفية أو شبكات بيانية خاصة بالاتصالات الحاسوبية .

وكما هو معلوم فإن المؤسسات الكبرى عندما استخدمت أجهزة الحواسيب على نطاق واسع في هذا القرن لغرض تبادل المعلومات ومعالجة البيانات تطوّر كذلك معها عملية تناقل المعلومات فيما بين هذه المؤسسات حيث كان النموذج القائم هو نموذج أجهزة الكمبيوتر المركزية والمحطات الطرفية بحيث يقوم المستخدم عادة بإرسال واستقبال المعلومات التي يحتاجها إلى ومن الجهاز المركزي والذي يقوم بمعالجة كافة ما يرد إليه من بيانات ، وبناءً على ذلك قامت صناعة كاملة هي صناعة الأجهزة الايوانية (الكمبيوتر الجبار) والتي كانت بالنسبة لزمناها ذات قدرات معالجة خارقة ، ومع أن هذا النموذج كان يسمح للمئات وفي بعض الأحيان للآلاف من المستخدمين في مشاركة المعلومات و استخدام الكمبيوتر إلا

أن اكبر معوقات هذا الأسلوب كان ثمن الأجهزة المرتفعة وكذلك بطء الاتصالات وصعوبة الاستخدام ، وبانتشار الحاسوب الشخصي الرخيص الثمن والسريع أخذت المؤسسات تبني نموذجاً جديداً في بناء الاتصالات فيما بينها من خلال بناء الشبكات التي تقوم على مجموعة جديدة من نظم التشغيل التي تتميز نسبياً أيضاً بسهولة الاستخدام ، وفي داخل المؤسسات أيضاً، اعتمدت النماذج المركزية لإدارة هذه الشبكات بحيث يتم ربطها بالأجهزة الأيوانية أو الحواسيب الصغيرة لغرض الاتصال مع الغير أو الاتصال فيما بين المستخدمين أنفسهم لغرض تنقل المعلومات والبيانات المخزنة والاستفادة منها ، وفي خلال التطور الحاصل في نقل البيانات والمعلومات والاتصال مع الغير ، جاء عقد التسعينات حاملاً معه ما عرف أنه أكبر ثورة في نقل المعلومات من خلال الأنترنت والتي استطاعت أن تخلق اتصالاً مع الغير ومع الآخرين والتي ربطت الملايين من أجهزة الحواسيب وساعدت في نقل البيانات والمعلومات وكونت بيئة مؤتمته جديدة تسمح بالمطورين ببناء تطبيقات جديدة تماماً ، من حيث إمكانية الفهم والتعامل مع مختلف الحواسيب دون الحاجة إلى عمليات تحويل من أية نوع كان ، ومع ظهور الأنترنت ظهرت نماذج جديدة للحوسبة مثل جافا (Jafa) وهي لغة برمجة معقدة تمكن من صنع برامج تعمل في أي بيئة تشغيل تدعمها وبهذا تم التوصل إلى ما يعرف الآن بتنقل المعلومات والاتصال مع الغير بسهولة ويسر .

(1 - 2) نبذة تعريفية عن الحاسوب

(1 - 2 - 1) المفاهيم الأساسية للحاسوب

شهدت السنوات الأخيرة انتشاراً واسعاً لاستخدامات الحاسوب في جميع مناحي الحياة، فقد فرض الحاسوب نفسه بما يمتلك من الإمكانيات الهائلة على سائر العلوم ، وذلك بسبب تزايد حجوم البيانات والمعلومات المرافقة للتطور العلمي والتقني المستمر في مختلف الميادين ، هذا التزايد ألزم الإنسان اللجوء إلى نظم المعالجة

الآلية الحديثة ، بدلاً من الأنظمة اليدوية، والمتمثلة بالأنظمة الحاسوبية التي أصبحت في وقتنا الحالي أداة فعالة وأساسية في حفظ ومعالجة البيانات مهما يكن حجمها وبالسرية القصوى .

ولكي نستطيع تسخير إمكانيات الحاسوب في أي مجال من مجالات الحياة بهدف مواكبة التطور العلمي والتكنولوجي لا بد من توفر عناصر أخرى مساندة ومدعمة لعمل الحاسوب كجهاز ، وهذا ما استدعي إلى إنشاء نظام حاسوبي الذي يتألف من العناصر الأساسية التالية :

أولاً : الأجهزة Hardware

وهي عبارة عن المكونات المادية والتي تشتمل على جهاز الحاسوب والأجهزة المساعدة له . إن التطور السريع في عالم الحواسيب أدى إلى إفراز تشكيلة واسعة ومتنوعة من الحواسيب تتراوح من حواسيب مايكروية زهيدة الثمن إلى أخرى كبيرة ومعقدة غالية الثمن ، إلا أن جميع الحواسيب على اختلاف حجمها وتصنيفها والمهام التي يكمن أن تؤديها ، تشترك في عدد من الخصائص والمواصفات، إذ أن جميعها تتكون من الوحدات التالية انظر الشكل (1) :

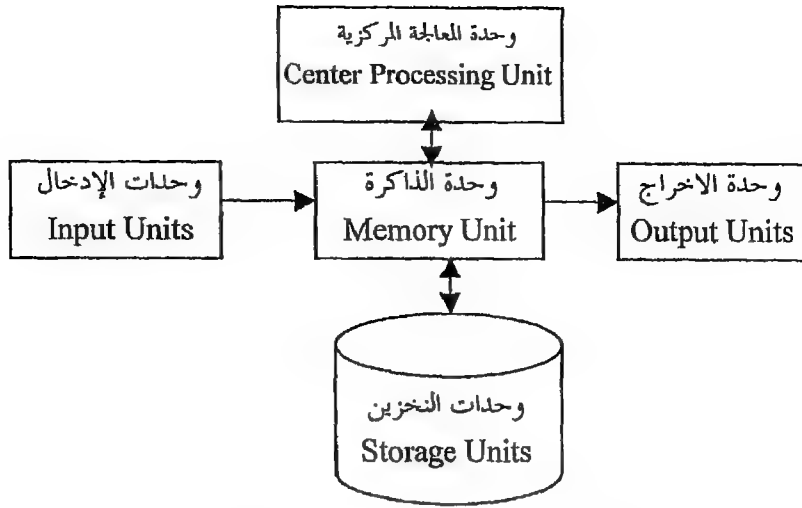
1 - وحدة المعالجة Central Processing Unit

2 - الذاكرة Memory

3 - وحدات الإدخال والإخراج Input / Output Units

4 - وحدات التخزين المساعدة Storage Units

و سنأتي على شرح المكونات المادية إن شاء الله في فصول لاحقة .



الشكل (1) الهيكل العام لمكونات الحاسوب

ثانياً : البرمجيات Software

هي مجموعة الأوامر أو التعليمات المتسلسلة التنفيذ و التي تستخدم في إدارة ومراقبة وتشغيل جهاز الحاسوب ، هذه الأوامر والتعليمات تسمى البرنامج (Program) وتصنف عادة البرمجيات إلى نوعين :

1. برمجيات النظم System Software :

وهي مجموعة البرامج تساعد في تشغيل نظام الحاسوب وتعمل على توجيه أجهزة الحاسوب المختلفة لاداء الوظائف الأساسية بها عند تنفيذ برنامج المستخدم بشكل تلقائي.

2. البرمجيات التطبيقية Application Software :

وهي البرامج المعدة لخدمة المستخدم في إحدى المجالات مثل برنامج (الوورد) (Word) (اكسل Excel) (برامج المحاسبة) (برامج تحليل المبيعات ومراقبة المخزون) (برامج الرسم الهندسي) ...

وسنتعرف على المزيد من البرمجيات عندما تتناول فصل البرمجيات بشكل مفصل إن شاء الله .

ثالثاً : البيانات Data

وهي مفاهيم لغوية أو رياضية أو رمزية متفق عليها لتمثل الأحداث أو الأشخاص أو الأشياء ، هذه المفاهيم تحتاج إلى معالجة Processing كي تتحول إلى معلومات مفيدة لمن يحتاجها ، والحاسوب لديه القدرة على هذه المعالجة بسرعة كبيرة ودقة متناهية مهما يكن حجم البيانات المراد معالجتها ، وهذا ما أعطى للحاسوب هذه المكانة المرموقة بين سائر العلوم .

رابعاً : العنصر البشري Users

وهم مستخدمي نظام الحاسوب والمستخدمون من تشغيله ويثقف نجاح نظام الحاسوب وقدرته إلى حد كبير على كفاءة العنصر البشري وتمكنه من استخدام الحاسوب وقدرته على تسخير هذا الجهاز بواسطة البرمجيات المتاحة والبيانات المتوفرة في تحقيق أهدافه، وبسبب توسع وتطور علم الحاسوب أصبح هناك متخصصين في شتى أقسامه وبالتالي يكن تصنيف العنصر البشري إلى فئات مثل :

مبرمج Programmer ، محلل نظم Processing Analysis

مهندس برمجيات Software Engineer ،

الذي يقوم بإدخال البيانات Typist ، مشغل الحاسوب Operator ،

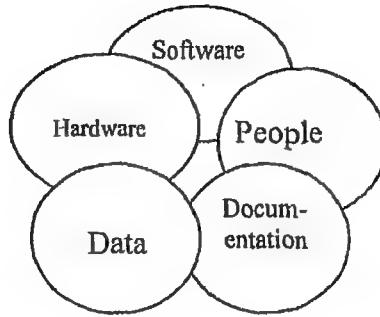
الفني الذي يقوم بفحص أجزاء الحاسوب وتغيير الأجزاء المعطلة .

خامساً : التعليمات والوثائق Documentation

و هي مجموعة التعليمات الموثقة والتي توضح للمستخدم كيفية استخدام النظام الحاسوبي وتشغيله لتنفيذ التطبيقات المختلفة المطلوبة،

والإجراءات اللازمة اتباعها عند حدوث أي عطل أو خلل في أداء النظام الحاسوبي .

إن العناصر السابقة ترتبط بشكل وثيق مع بعضها البعض وأي قصور في أي عنصر منها يؤدي إلى قصور عمل النظام الحاسوبي ، ويمكن تمثيل الترابط بين العناصر كما في الشكل (2) .



الشكل (2)

(1 - 2 - 2) تعريف الحاسوب وخصائصه

إن كلمة كمبيوتر (Computer) مشتقة من فعل يحسب Compute ، لذلك عرّبت كلمة كمبيوتر في اللغات الأجنبية إلى كلمة حاسوب في اللغة العربية ويعرف الحاسوب بتعاريف مختلفة نذكر منها :

- الحاسوب : هو آلة حاسبة إلكترونية ذات سرعة عالية جداً ، ودقة متناهية يمكنها قبول البيانات وتخزينها ومعالجتها للوصول إلى النتائج المطلوبة .

هو نظام إلكتروني لمعالجة البيانات وفقاً لمجموعة من التعليمات .

- هو مجموعة من الأجهزة الإلكترونية تدعى Hardware يتم التحكم في أدائها بواسطة مجموعة من البرامج المخزنة تدعى Software .

ومن هذه التعريفات واستناداً إلى عمل النظام الحاسوبي يتضح لنا أن الحاسوب ليس مجرد جهاز أو آلة ولكنه نظام متكامل يتكون من مجموعة الأجهزة

Hardware و مجموعة التعليمات المخزنة Software ، مدخلات هذا النظام هي البيانات Data التي يقوم الحاسوب بمعالجتها أو تخزينها أو كليهما ، أما مخرجاته فهي المعلومات Information أو النتائج Results . ولا يتم ذلك كله بمعزل عن الإنسان المشغل والمستخدم والمستثمر لهذا النظام، أي يتم تحت إشرافه وإدارته ومراقبته وتخطيطه الكامل .

ويتميز الحاسوب بإمكانيات وقدرات خاصة نذكر منها :

1. السرعة الفائقة في أداء و تنفيذ التعليمات :

يقوم الحاسوب بالإضافة إلى التعليمات الحسابية المعروفة ، بمقارنة الحروف حسب تسلسلها الهجائي وكذلك تحرير ونقل ونسخ وتخزين الأرقام والحروف والرموز والكلمات بسرعة فائقة ، إذ أن الحاسوب الصغير يمكنه تنفيذ مئات الآلاف من التعليمات في الثانية ، بينما الحاسوب الأكبر يمكنه تنفيذ عدة ملايين، لذلك أصبح بالإمكان في هذه الأيام من معالجة كمية هائلة من البيانات والحصول على كم من المعلومات ، كان من العسير الحصول عليها منذ سنوات قليلة مضت إلا بعد مجهود بشري كبير يستغرق وقتاً طويلاً.

2. الدقة في تنفيذ العمليات المختلفة :

بالإضافة إلى السرعة الفائقة فإن الحاسوب يعطي النتائج بدقة متناهية، إذا كانت البيانات المدخلة صحيحة وإذا كان البرنامج الذي يقوم بمعالجة البيانات صحيحاً ومنطقياً ، وكلاهما البيانات المدخلة والبرنامج من صنع الإنسان فإذا كان هناك قصوراً في أحدهما فإن النتائج ستكون غير صحيحة .

3. القدرة على العمل فترات طويلة دون أخطاء :

يتكون الحاسوب من دوائر إلكترونية، وهي ليست عرضة للتآكل مثل المكونات الميكانيكية ، ونظراً للتطور الهائل في تكنولوجيا تصنيع الإلكترونيات، فإن الدوائر الإلكترونية أصبحت ذات كفاءة عالية جداً ، إذ بإمكانها أداء مئات الآلاف

أو الملايين من العمليات كل ثانية ولفترات طويلة دون تعب أو ملل فإذا كان هناك 5 ملايين عملية حسابية يجب إجرائها فإن الحاسوب ينفذ العملية رقم الـ 5 مليون بنفس الدقة والسرعة تماماً كما ينفذ العملية الأولى .

4. تعدد الاستعمال :

حيث يستطيع الحاسوب حل كثير من المشاكل التي تواجه الإنسان من خلال أنواع متعددة من البرامج مثل برامج معالجة النصوص ، برامج النوافذ ، برامج قواعد البيانات...

5. الكفاءة العالية في إدارة البيانات :

بمجرد إدخال البيانات وتشغيلها يقوم الحاسوب بتنفيذ أحد أو بعض أو كل العمليات التالية :

- التخزين لحفظ البيانات والمعلومات لحين الحاجة إليها .
- الاسترجاع أي استعادة البيانات والمعلومات المخزنة لإعادة استخدامها أو الاطلاع عليها عند الحاجة .
- نقل المعلومات من موقع إلى آخر عبر قنوات اتصال لاستخدامها أو لإجراء المزيد من عمليات التشغيل والمعالجة لها حتى تصبح في الصورة المطلوبة للمستخدم النهائي .
- نسخ المعلومات أو إعادة إنتاجها ، لإنشاء نسخة احتياطية من البيانات والمعلومات .

6. تنفيذ العمليات المطلوبة آلياً :

يقوم الحاسوب عادة عن طريق البرنامج المصمم له بأداء عمليات متعددة دون تدخل الإنسان ، إذ أن تركيب البرنامج وهيكلته تساعد الحاسوب على الانتقال بين التعليمات بسرعة كبيرة وبصورة آلية .
رغم الخصائص والميزات السابقة للحاسوب ، هناك عدداً من الأمور تحدد

من إمكانيات وقدرات الحاسوب منها :

1- انعدام الذكاء الفطري :

وذلك أن الحاسوب لا يفكر، وينفذ ما يطلب منه من أوامر وتعليمات غير مذكورة في البرنامج.

2- تتوقف كفاءته على كفاءة ودقة البرامج :

لأنه كما ذكرنا لا يمكن للحاسوب التفكير وبالتالي كلما كانت البرامج التي تضم التعليمات والأوامر دقيقة ومنطقية، كلما كان عمل الحاسوب أفضل، أضف إلى ذلك فإن البرنامج الجيد لا يعطي النتائج المطلوبة إلا إذا تم تغذية الحاسوب بالبيانات اللازمة والصحيحة والموثوقة .

3- النظام الحاسوبي أقل مرونة من النظام اليدوي وذلك لأن أي خلل في أي عنصر من عناصره يتطلب وقتاً وجهداً لاصلاحه .

(3-2-1) أنواع الحواسيب

إن ازدياد الحاجة إلى استخدام الحواسيب في شتى المجالات وتنوع استخدامها، أدى إلى ظهور أنواع مختلفة منها ، إذ ظهرت الحواسيب المترتبة والحواسيب المكتبية والحواسيب للاستخدامات الإدارية والحواسيب لاستخدامات مراكز البحوث ومراقبة الإنتاج والتحكم به . وغيرها من الأجهزة ، أضف إلى ذلك اختلفت تلك الأنواع بحجمها واتساع ذاكرتها وطريقة معالجة البيانات. لذلك يمكن تصنيف الحواسيب إلى ثلاثة أنواع من حيث العمل :

النوع الأول : حسب أسلوب معالجة البيانات :

وتصنف إلى ثلاثة أنواع أيضاً :

1- حواسيب رقمية Digital Computers :

هذا النوع من الحواسيب يعالج البيانات بتحويلها إلى أرقام حسب نظام

عددي معين، ثم يقوم بمعالجتها حسابياً ومنطقياً ومن ثم تحويل الأرقام المعالجة إلى المعلومات المطلوبة، وتستخدم الحواسيب الرقمية في المجالات العلمية والتجارية والاجتماعية.

2- حواسيب قياسية Analog Computers :

هذا النوع يعالج البيانات باستخدام طرق القياس التي تعتمد على الظواهر الفيزيائية مثل الضغط الجوي ودرجة الحرارة و الجهد الكهربائي،.. وهذا النوع من الحواسيب تستخدم في مجال الدراسات العلمية و مجالات التحكم الآلي.

3- حواسيب مهجنة Hybrid Computers :

هذه الحواسيب تجمع بين خصائص الحواسيب الرقمية و الحواسيب القياسية وتستخدم في بعض المجالات العلمية الخاصة مثل أبحاث الفضاء و الاستشعار عن بعد.

النوع الثاني : حسب الغرض من استخدامها :

و تصنف إلى نوعين :

1- حواسيب ذات الغرض الخاص Special- Purpose Computers

و هي حواسيب مصممة لتنفيذ غرض معين والقيام بوظيفة محددة مثل الحواسيب الخاصة بالطيران الآلي - الحواسيب المستخدمة لأغراض عسكرية كتوجيه الصواريخ والتحكم بشبكات الرادار - الحواسيب المستخدمة في الصناعة للتحكم بخطوط الإنتاج

2- حواسيب ذات الغرض العام General-Purpose Computers

وهي حواسيب مصممة لتنفيذ العديد من المهام في المجالات العلمية المختلفة والأعمال الإدارية ، وذلك باستخدام برمجيات مختلفة حسب نوع المشكلة المراد معالجتها .

النوع الثالث : حسب حجم الذاكرة :

وتصنف إلى خمسة مجموعات :

1. الحواسيب العملاقة : Super Computers

تتميز بوجود ذاكرة ضخمة ، سرعتها كبيرة جداً و لها اكثر من معالج واحد . استخدامها محدود جداً بسبب كلفتها الباهظة .

في عام 1981 أعلنت مجموعة من الشركات المنتجة للحواسيب العملاقة عن إنتاج حاسوب أسموه (الحاسوب الميكرو العملاق Supermicro Computer ، حيث قدراته تقترب من الحاسوب العملاق و ذاكرته تزيد عن 128 ميجابايت .

2. الحواسيب الكبيرة Mainframe Computers

تتميز بوجود ذاكرة رئيسية كبيرة يمكن توسيعها ، وتستخدم مع هذه الحواسيب أجهزة طرفية تصل إلى مئات من الأجهزة ، بالإضافة إلى أن سرعة وحدة المعالجة المركزية كبيرة ، لذلك تستخدم بشكل واسع في الشركات الكبيرة مثل شركات الطيران و إدارات الهجرة و الجوازات وهيئات الدفاع...
إلا أن مساوئها كثيرة منها أنها تحتاج إلى مصدر خاص للطاقة و نظام للتهوية و تحتاج إلى عدد من الموظفين لتشغيلها و إدارتها ذوي خبرة واختصاص في الحواسيب الكبيرة بالإضافة إلى ارتفاع ثمنها .

3. حواسيب الميني Mini Computers

أهم مزاياها وجودها في صورة عائلة بحيث يمكن للمستثمر استبدال حواسيبه بحواسيب أعلى من نفس العائلة و بالسعر المناسب كلما دعت الحاجة، بالإضافة إلى أنها تتيح تعدد المستخدمين يصل إلى مائة مستخدم أو أكثر ، لذلك تستخدم بشكل واسع في المؤسسات والمنظمات التي تتطلب قدرات كبيرة في تخزين البيانات و معالجتها إلا أن سعرها مرتفع .

4. حواسيب الميكرو Micro Computers

وهي أصغر الحواسيب حجماً وأقلها تكلفة و يوجد في الأسواق الآن العديد من الأنواع المختلفة حسب حجم ذاكرتها و نوع معالجها وسرعة أداء العمليات ، نذكر منها :

- الحواسيب المنزلية Home Compute

- الحواسيب المحمولة Portable Computers

- الحواسيب المكتبية Desktop Computers

- الحواسيب الشخصية Personal Computers

و تعتبر الحواسيب الشخصية أكثر الحواسيب انتشاراً ، تستخدم في مجالات الأعمال الصغيرة والمتوسطة بالإضافة إلى استخداماتها الشخصية المختلفة، وذلك لسهولة استخدامها و توفر البرمجيات التطبيقية لها .

نشير هنا إلى أن هذه الحواسيب في تطوير مستمر من حيث حجم ذاكرتها و سرعة معالجها وأسلوب التعامل معها .

(1 - 2 - 4) استخدامات الحواسيب

في الوقت الراهن يقاس مدى تطور وتقدم أي مؤسسة أو شركة بمدى استخدامها للحاسوب و بحجم وسرعة الخدمات التي تنجزها أجهزتها الحاسوبية ، إذ أن الحاسوب دخل جميع ميادين العلم و المعرفة بسبب الحاجة الماسة لدى جميع الأوساط إلى الحصول على نتائج متناهية في الدقة و سرعة في الإنجاز و المعالجة للبيانات الكثيرة و المتزايدة يوماً بعد يوم ، حتى أنه أصبح من الصعب أن نجد اليوم علماً من العلوم لم يتدخل الحاسوب فيه، ومن الصعب أن نحصر جميع الاستخدامات البشرية لهذا الحاسوب المتميز بالإمكانات والقدرات ، إلا أننا سنوجز بعض الاستخدامات الشائعة له :

- الشؤون الإدارية و تنظيم ملفات الموظفين و المعدات و حفظ جميع البيانات الإدارية المتعلقة بالدوائر .
- الدراسات المالية و المحاسبية و الرواتب .
- المبيعات و التسويق و الترويج على البضائع و لاسيما حاليا من خلال شبكة الأنترنت .
- الدراسات الإحصائية و ذلك لمعالجة البيانات الإحصائية أثناء عمليات التعداد السكاني و عمليات المسح الجغرافي و الديموغرافي بالإضافة إلى تحليل البيانات و استخدامها في مجالات التنبؤ و تهيئة المعلومات اللازمة للدراسات الاقتصادية بمختلف أنواعها .
- أعمال السكرتارية و الطباعة ، و تهيئة المستندات و العقود اللازمة بشكل فريد من نوعه، و اعداد الكتب و المراجع العلمية و اخراجها بشكل لائق .
- في تنظيم أعمال المخازن و المستودعات و تهيئة الطلبات اللازمة للاستيراد و التصدير .
- اتمتة المكاتب و البريد الإلكتروني .
- أعمال البنوك و البورصات المالية .
- في مجالات تخطيط الإنتاج و الرقابة عليه باستخدام طرق علمية وأساليب مدروسة .
- في مجالات تنظيم السفر و حجز أماكن المسافرين على الطائرات و القطارات و الباصات ... و تهيئة البيانات حول موعد وصول أو مغادرة وسائل النقل .
- في مجالات العلوم الطبية فقد صار بإمكان الحاسوب التحكم بغرف العمليات و أجهزة التصوير المختلفة و الأجهزة الطبية المنظورة و تهيئة جميع البيانات المتعلقة بالمشكلة المعالجة، بالإضافة إلى ربط المستشفيات المعالجة مع بعضها لخدمة الطب و الاستشارات الطبية و تبادل الخبرات الطبية .

- في مجالات الهندسة فقد أصبح الحاسوب الوسيلة الأساسية في إجراء الحسابات اللازمة لإنشاء وبناء المشاريع الهندسية ، وفي وضع التصميم الهندسية والتحكم بها إذ بالإمكان إنشاء العديد من التصميم لاختيار أفضلها ...

- في العلوم العسكرية و خاصة في دراسة الإستراتيجيات العسكرية للجيش وتحديد مواقع وإمكانات العدو من خلال تحليل كم من البيانات وإعطاء الحلول المثلى بسرعة و دقة فائقة معتمدا على العلوم العسكرية - الرياضية و من أهمها نظرية الألعاب الإستراتيجية Games theory .

- في مجالات التعليم فقد ساهم و يساهم في تطوير التعليم من خلال برامج تعليمية موجهة للأطفال و الكبار ، و من خلال مساعدة الطلبة و المدرسين على التروء بالمعرفة و من خلال الخدمات العلمية التي يقدمها عبر شبكات الأنترنت .بالإضافة إلى كونه أصبح أداة لتطوير و تحديث الكثير من الأساليب و الطرق الرياضية و الفيزيائية و الكيميائية .

- في مجال الفلك فقد أصبح الحاسوب يتحكم في توجيه الأقمار الصناعية ورصد دوران الكواكب و المجرات و اكتشاف المزيد من خفايا الكون ...

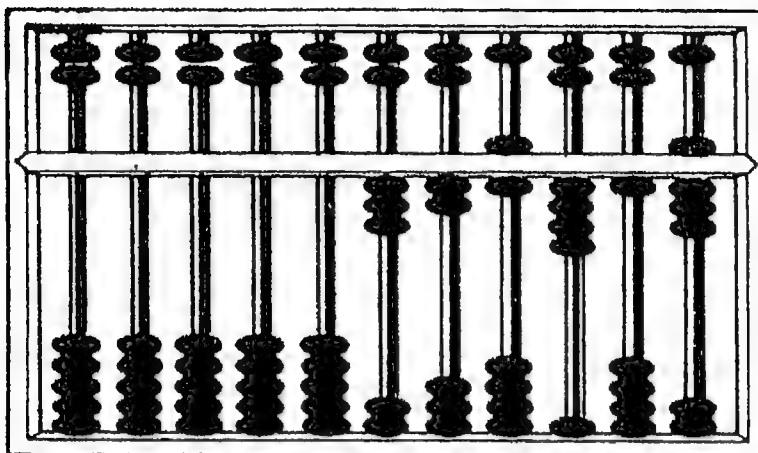
(1- 2 - 5) لمحة تاريخية عن تطور الحواسيب

إن اختراع الحاسوب لم يأتي من فراغ ، بل كان نتيجة لمحاولات و أفكار عديدة سابقة بهدف تسهيل إجراء الحسابات وتطوير وسائل معالجة البيانات التي صار يزداد حجمها يوماً بعد يوم، لذلك سنعرض في هذه الفقرة الأفكار والآلات التي أدت وكانت السبب في وصول الحاسوب إلى ما هو عليه في الوقت الحاضر .

1- استخدام الحصى لمعرفة كمية الأشياء ثم استعمال أصابع اليد في العد .

2- Abacus :هو أول وسائل الحساب اليدوي أستخدم لإجراء العمليات الحسابية الأربع ، و يتألف من إطار خشبي يحوي عدداً من الأعمدة، كل عامود يضم عدد من الكرات أو الخرزات مستديرة الشكل على الأعمدة.

الشكل (3) ، يعود اكتشاف العداد إلى عام 2000 ق.م في كل من الصين واليابان و روسيا ، واستخدم خلال الحضارات القديمة ، واستمر مسيطراً مئات من السنين ، فقد شاع استخدامه في أوروبا في القرن السادس عشر الميلادي ، ومازال مستخدماً حتى الآن في بعض البلدان مثل روسيا و اليابان .



الشكل (3) أحد أشكال العداد

3- أعمدة نابير Napier's Roods and Bones

وضع هذه الوسيلة العالم الرياضي جون نابير (John Napier) عام (1617) وهي عبارة عن جداول رياضية تعمل على أساس اللوغاريتمات لتحويل عملية الضرب إلى عملية جمع وفق أسلوب معين ، صنعت على هيئة شرائح خشبية كل منها مقسمة إلى تسعة مربعات صغيرة وكل مربع يقسم إلى قسمين ، كل عامود يمثل جدول الضرب للأرقام من 1 وحتى 9 بحيث يكتب رقم الآحاد أسفل قطر المربع أما رقم العشرات فيكتب أعلى القطر الرئيسي الشكل (4).

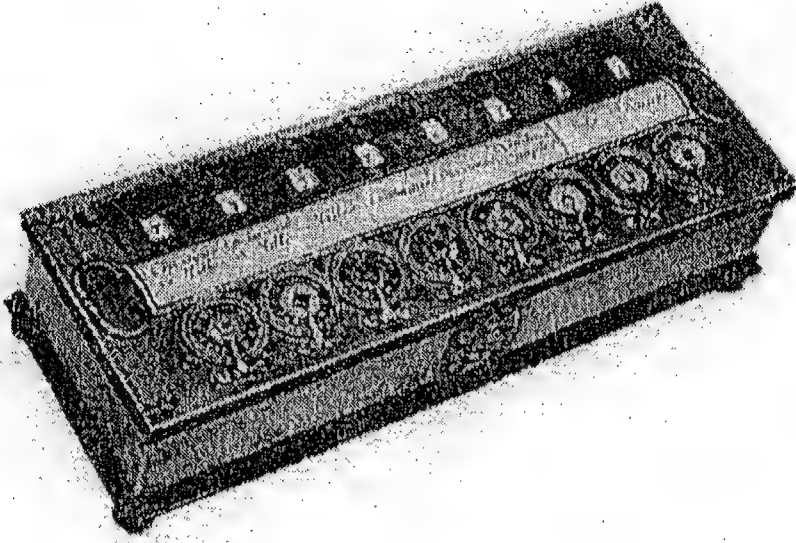
✓	✓	✓	✓							
1	2	3	4	5	6	7	8	9	0	
2	4	6	8	1	0	1	2	4	6	8
3	6	9	2	1	5	1	8	2	4	7
4	8	1	1	2	0	2	4	3	2	3
5	1	1	2	2	3	3	4	4	0	4
6	1	1	2	3	3	4	4	5	0	4
7	1	2	2	3	4	4	5	6	3	0
8	1	2	3	4	4	5	6	7	2	0
9	1	2	3	4	5	6	7	8	1	0

الشكل (4) أعمدة نابير

4- آلة باسكال Pascal Machine

وهي آلة ميكانيكية للجمع و الطرح اخترعها العالم الرياضي الفرنسي بلير باسكال (Plaise Pascal) عام 1642 بهدف تسهيل عملية مسك الدفاتر المحاسبية في جمع الضرائب لوالده . والآلة عبارة عن مجموعة عجلات كل عجلة مقسم إلى عشرة مقاطع مسجل عليها الأرقام من 1 إلى 9 وتتصل العجلات منع بعضها البعض بحيث تدور العجلة الثانية حركة واحدة عندما تدور العجلة الأولى دورة كاملة و هكذا بالنسبة لباقي العجلات، حيث أن العجلة الأولى تمثل منزلة الآحاد والعجلة الثانية تمثل منزلة العشرات وهكذا، الشكل (5)، تعتبر هذه الآلة الأساس في تصميم الآلات الميكانيكية التي ظهرت فيما بعد و حتى منتصف

الستينات، وتكرّما لاسم العالم باسكال أطلق اسمه على إحدى لغات برمجة الحاسوب (لغة باسكال) .



الشكل (5) آلة باسكال

5- آلة ليبنيز (Leibniz Machine)

تعتبر هذه الآلة تطوير لآلة باسكال تقوم بعمليات الضرب و القسمة واستخراج الجذور بالإضافة إلى عمليات الجمع والطرح، صممها العالم الألماني جوتفريد وليام ليبنيز Gottfried Wilhelm Leibniz عام 1671 .

6- آلة جون ميلر J. H. Muller حيث اقترح العالم جون ميلر 1786 ببناء حاسبة تقوم بحساب الدوال الرياضية بشكل تلقائي.

7- آلة الفروقات Difference Machine

بنى هذه الآلة العالم الإنجليزي شارل بابليج Charles Babbage عام 1922 اعتماداً على فكرة العالم ميلر لحساب الدوال الرياضية بشكل تلقائي و بناء الجداول اللوغريتمات باستخدام طريقة الفروقات و سميت بآلة الفروقات إلا أن هذه الآلة بنيت بشكل جزئي بسبب التكاليف المالية الباهظة .

8- الآلة التحليلية Analytical Machine

صممها العالم باباج عام 1937 على نفقته الخاصة ، وهي آلة متطورة تحتوي الكثير من مواصفات الحاسوب الرقمي العصري مثل وحدة الذاكرة ووحدات الإدخال والإخراج .

9- نظام البطاقات المثقبة

يعتبر العالم هرمان هوليرث Herman Hollerith (1860-1929) أول من وضع نظام البطاقات المثقبة لتخزين البيانات ضمن قواعد محددة، حيث صمم آلة لترتيب البطاقات و آلة لوضع البيانات في جداول .

10- الآلات الكهروميكانيكية Electromechanical Machine

لم تتوقف الأبحاث و النظريات حول تصميم أجهزة تخزين ومعالجة البيانات سوى أثناء الحرب العالمية الثانية ، بعدها قام العالم الأمريكي هوارد ايكن Howard Aicon الأستاذ في جامعة هارفارد بالولايات المتحدة الأمريكية بتصميم أول حاسوب كهر وميكانيكي معتمداً على أفكار وتصاميم العالم الإنجليزي باباج وقامت شركة IBM الأمريكية بتنفيذ هذا التصميم علم 1944 و سمي بمارك 1- (Mark-1) أهم ميزاته :

1. طوله 51 قدما .
2. عدد مفاتيحه 300 مفتاح .
3. ارتفاعه 8 أقدام .
4. طول أسلاكه 500 ميل .
5. عدد قطعه اكثر من مليون .
6. بطيء في سرعة العمليات الحسابية .
7. طريقة تشغيله معقدة جدا إذ بلغ عدد صفحات دفتر التعليمات الخاصة بتشغيله (547) صفحة يجب إتقانها من قبل كل مستخدم.
8. يستقبل البيانات على بطاقات مثقبة .
9. يقوم بالعمليات الأربع و حساب اللوغريتمات و النسب المثلثية .

11- الآلات الإلكترونية Electronic Machines

- في عام 1946 قام العالمان ايكارت J. P. Echert و موشلي J. W. Mauchly الأستاذان في كلية مور للهندسة في جامعة بنسلفانيا بالولايات المتحدة الأمريكية بتصميم أول حاسوب إلكتروني رقمي باسم: Electronic Numerical Integrator and) ENIAC (Automatic Calculators) وقامت شركة بل للتليفونات (Bell Telephone Co.) بتنفيذه ، في صناعته 18 ألف أنبوب مفرغ تعمل على أساس النبضات الكهربائية ، و قد استخدم هذا الجهاز في التعداد العام للسكان في الولايات المتحدة الأمريكية إذ كان لهذا الاستخدام الأثر الكبير في مجال هيئة البيانات الإحصائية .

-خلال الأعوام (1946 – 1949)

اقترح العالم الأمريكي (J. V. Neuman) بناء حاسوب يخزن البيانات والتعليمات سرية على شكل شيفرات معينة ، ودعيت فكرته بمبدأ تخزين البرامج وطبقت فكرته على الحاسوب Electronic : EDVAC Discrete Variable Automatic Computer ، ثم قام العالم ويلكس (M.V.Wilkes) من جامعة كامبرج - إنجلترا بتطوير الحاسوب السابق و أسموه (EDSAC) (Electronic Delay Storage Automatic Computer) حيث استخدم فيه ذاكرة لحفظ البرامج .

-عام 1952 ظهر حاسوب باسم UNIVAC (Universal Automatic Computer) لخدمة معالجة البيانات الإدارية مستخدماً أنابيب كهربائية مفرغة ، و هو أول حاسوب إلكتروني ظهر على المستوى التجاري بهدف التسويق من قبل شركة راند .

-بعد ذلك انتشرت صناعة الحواسيب الإلكترونية بشكل واسع فظهرت حاسبات شركة (IBM) وشركة (NCR) وشركة (ICL) الانكليزية وغيرها من الشركات .

وحتى يومنا هذا ما زال السباق على قدم وساق في عالم تطوير الحواسيب واستخداماتها .

(1 - 2 - 6) أجيال الحواسيب Computer Generations

نتيجة للتطور المستمر لأجهزة الحواسيب منذ عام 1944 (منذ اختراع أول حاسوب (Mark - 1)) وحتى يومنا هذا ، فقد صنف المختصون في عالم الحوسبة أجهزة الحواسيب ضمن فترات زمنية دعوها الأجيال Generations .

أولا : حواسيب الجيل الأول First Generation Computers

وتشمل على الحواسيب التي أنتجت خلال الفترة الزمنية 1945 - 1959 وتدعى حواسيب الأنابيب المفرغة . وقد تأثرت حواسيب هذا الجيل بالأعمال البحثية الرائدة والتي نذكر منها ثلاثة أعمال هامة :

العمل الأول : للباحث آلان تورينج Alan Turning الذي صمم أول نموذج للآلات المنطقية عام 1926 و صمم أيضا أول اختبار لذكاء الآلات الذي يطلق عليه اختبار تورينج Turning test .

العمل الثاني : للباحثان الفرنسيان كوفيغنال (Couffignal) و فالتان (Valtan) اللذان نشرتا في عام 1936 بحثا عن بعض الأفكار النظرية لتطبيق النظام الثنائي في بناء الحاسوب .

العمل الثالث : فيعود للعالم الرياضي جون فون نيومان (J.V.Neuman) وآخرون في توثيق المبادئ الأساسية للحاسبات الرقمية عام 1945 من خلال مسودة تقرير عن الحاسوب EDVAC ، حيث وضحوا في التقرير إمكانية استخدام النظام الثنائي في بناء الحاسبات و طريقة

استخدام الدوائر الرقمية على التوازي في الوحدات الحسابية و
اختبارها ، وإمكانية تخزين البرامج و البيانات في ذاكرة الحاسوب .

ومنذ عام 1938 وحتى عام 1959 أنتجت العديد من الحواسيب المختلفة
في ألمانيا وأمريكا وبريطانيا .. تلبية حاجة مخترعيهم لأغراض مختلفة منها الحواسيب
. ENIAC , EDVAC , EDSAC , UNIVAS

و من أبرز الخصائص و الملامح التي تتميز بها حاسبات الجيل الأول هي :

1. الاعتماد على تكنولوجيا الصمامات المفرغة Vacuum Tubes ، مما أدى إلى

ضخامة المعدات و كبر حجمها ، و استهلاكها طاقة كهربائية كبيرة .

2. الحرارة الناتجة عن تشغيل الحاسبات تزداد مع طيلة فترة الاستخدام مما يتطلب

تهيئة المناخ المناسب من حيث التبريد و الرطوبة لضمان استمرارية العمل ،

وإلا يجب إيقاف الأجهزة عن العمل بين الحين و الآخر .

3. سرعة إنجاز العمليات بطيئة (حيث لا يتجاوز 1000 عملية في الثانية) ، فيما

إذا قورنت بحاسبات الأجيال اللاحقة بالإضافة إلى سعة ذاكرتها المتواضعة.

4. غالية الثمن مما اضطر الناس إلى استئجارها بدلا من شرائها .

5. استخدم في برمجة حواسيب هذا الجيل لغة الآلة Machine Language ، ثم

استخدم لغة التجميع Assembly Language في نهاية هذا الجيل .

6. استخدمت حواسيب هذا الجيل في عملها أسلوب الذاكرة المعنونة ، حيث

اعتمد عمل الذاكرة في بادئ الأمر على أشعة الكاثود (Cathode Ray) ثم

استبدلت بالذاكرة ذات الخلايا المغنطة

7. أجهزة الجيل الأول كانت تعمل على أساس نظام البرنامج الواحد (أي

تنفيذ برنامج واحد في كل مرة) .

8. استخدمت البطاقات المثقبة و الأشرطة الورقية المثقبة في تخزين البيانات .

ثانياً : حواسيب الجيل الثاني: Second Generation Computers:

وتشمل الحواسيب التي أنتجت من منتصف الخمسينات و حتى منتصف الستينات أي ضمن الفترة الزمنية 1959 و حتى 1964 وسميت بحواسيب الترانزستورات و قد تميزت حواسيب هذا الجيل بتطوراً كبيراً في مجال المكونات المادية Hardware و البرمجيات Software ففي مجال المكونات المادية :

1 - استخدام الترانزستور بدلاً من الأنابيب المفرغة (حيث مساحة الأنبوب المفرغ يمكن أن يحتوي على 200 ترانزستور) مما أدى إلى صغر حجم الحواسيب المنتجة و تقليل من استهلاك الطاقة الكهربائية اللازمة للتشغيل ، انعكس ذلك على التخفيف إلى حد ما من انتشار الحرارة الناتجة عن التشغيل.

2 - زيادة سرعة الأداء حيث أصبحت تقاس بالملي ثانية (1/1000 من الثانية).

أما في مجال البرمجيات :

1. بدأ استخدام اللغات ذات المستوى العالي (High Level Language) ومنها لغة الفورتران (Fortran) بالإضافة إلى استخدام لغة التجميع في برمجة الحواسيب .

2. بدأ استخدام الأقراص الممغنطة كوسيط لتخزين البيانات والمعلومات باستخدام أسلوب الوصول العشوائي ، هذا ما تم في الحاسوب (305 RMAC) حيث استخدم فيه 50 قرصاً ممغنطاً لتسجيل 5 مليون حرف مع إمكانية الوصول إلى أي سجل في زمن أقل من ثانية واحدة .

ومن الجدير بالذكر انه في نهاية هذا الجيل بدأ عدد من الشركات الصغيرة في إنتاج حواسيب قليلة التكلفة أطلق عليها حواسيب الميني

Low-Cost Minimal Computers (Mini)

ثالثاً : حواسيب الجيل الثالث : Third Generation Computers

- أطلق عليها أسم حواسيب الدارات المتكاملة وتدخل ضمن الفترة الزمنية 1964 وحتى عام 1971 وتعتبر عائلة الحواسيب (IBM / 300) من شركة IBM أول حواسيب هذا الجيل والتي استخدمت في المجالات العلمية والتجارية في آن واحد، من أهم الملامح والخصائص التي تميزت به حواسيب هذا الجيل :
- 1- استخدمت تكنولوجيا الدوائر المتكاملة (Integrated circuits) مما أدى إلى صغر حجم الحواسيب ووزنها ، وأصبحت تكلفتها أقل وكفاءتها أكبر في تنفيذ البرامج .
 - 2- استخدمت الذاكرات شبه الموصلة مما أدى إلى زيادة سعة الذاكرة وزيادة سرعة قراءة المعلومات من الذاكرة .
 - 3- زيادة سرعة الأداء حيث أنها أصبحت تقاس بالنانوثانية Nono second والتي تساوي إلى 10^9 من الثانية .
 - 4- استخدمت أنظمة التشغيل مما أدى إلى زيادة فعالية وأداء الأجهزة ، منها نظام البرمجة المتعددة Multiprogramming .
 - 5- ظهرت لغات برمجة جديدة لبرمجة الحواسيب .
 - 6- استخدمت أجهزة إدخال وإخراج جديدة مثل أجهزة القراءة الضوئية الشاشات الملونة .
 - 7- ظهور حواسيب الميني المتوسطة الحجم .

رابعاً : حواسيب الجيل الرابع Forth Generation Computers

والذي يعتبر بدايته عام 1971 مع إنتاج أول معالج باسم (4004) من شركة انتل Intel (المعالج هو عبارة عن شرائح إلكترونية متكاملة لعبت وما زالت تلعب دوراً هاماً في إنتاج الحواسيب ، سنأتي على شرح المعالجات في فصول لاحقة).

من أهم ميزات هذا الجيل :

1. انخفاض سعرها بسبب إنتاج حواسيب الميكروكمبيوتر Micro computer أحجامها صغيرة وكلفته قليلة وذات إمكانيات هائلة .
 2. ظهور المعالجات الميكروية Micro processors وميزاتها الفريدة .
 3. استمرار تطوير نظم التشغيل وظهور نظام تشغيل الأقراص Disk Operating System (DOS) للحواسيب الصغيرة .
 4. ظهور وسائط التخزين (الأقراص المرنة) Floppy Disk وظهور الراسمات وشاشات العرض المرئي .
 5. في مجال البرمجيات ظهور العديد من لغات البرمجة العالية المستوى منها الباسكال (Pascal) ولغة ال C والبرولوج Prolog .
 6. ظهور الحزم البرمجية التطبيقية المتكاملة Integrated software packages ، والتي تتيح للمستخدم متابعة تطبيقات متعددة في آن واحد عن طريق استخدام النوافذ Windows .
- وقد تعايش مع هذا الجيل انتشار شبكات الحاسوب على المستوى المحلي والعالمي وما تبع ذلك من تطور في نظم تشغيل الشبكات وتساعد مشاكل المواجهة بين أنواع الحواسيب المختلفة ومشاكل أمن البيانات .
- خامساً: حواسيب الجيل الخامس والأجيال المستقبلية

Fifth Generation Computers

يعتبر البعض أن هذا الجيل هو استمرارية للجيل الرابع ، والبعض يعتبر بداية التسعينات هو بداية جيل جديد من الحواسيب وذلك لظهور حواسيب متميزة ذات مواصفات هائلة وقد بدأ اليابانيون منذ منتصف الثمانينات في التخطيط لجيل الحواسيب الخامس التي تمتاز بالقدرة على فهم اللغات المحكية والمرسومة من خلال التعامل مع لغات قريبة من لغات الإنسان بالإضافة إلى سرعة معالجة كبيرة ، وفي عام 1992 تم إنتاج أول نموذج لهذا الجيل .

وقد أدى تطوير البرمجيات إلى ظهور النظام ويندوز Windows 3.0 عام 1990 الذي لم يكن نظام تشغيل قائم بحد ذاته فحسب ، بل شريحة برمجية تستند على النظام DOS . وفي عام 1991 طرحت شركة مايكروسوفت ويندوز 3.1 ثم ويندوز 3.11 (WINDOWS 3.11)، الذين وفرا دعماً أفضل لتكامل التطبيقات ووظائف السحب والإسقاط DRAG and DROP مما جعل هذا النظام مهيمناً على صناعة البرمجيات في مطلع التسعينات، بعد ذلك طور النظام ويندوز فظهر (Windows 95)، (Windows 97)، (Windows 98) مما أعطى هذا بعداً عملياً وفعالاً في مجال استخدامات الحاسوب في شتى المجالات .

أضف إلى ذلك فقد حققت لغتا فيجول بيسك Visual Basic وفيجول سي Visual c++ من مايكروسوفت انتصارات هامة في مجال اللغات وتطبيقاتها .

أما في مجال العتاد فقد تم تطوير المعالجات بشكل ملحوظ فعلى سبيل المثال طرحت شركة INTEL عام 1993 المعالج بنتيوم Pentium الذي احتوى على حوالي 3 مليون ترانزستور ، وبموازاة تقدم المعالجات تطورت أيضاً تقنيات الأقراص الصلبة والمرنة وسعات تخزينها وكذلك تقنيات عرض الرسومات على الشاشة ، فظهرت السرعات الرسومية لتحسين استجابة الشاشة مع تطبيقات ويندوز .

في مجال الأبحاث المتعلقة بتطوير الجيل الخامس والأجيال المستقبلية المتوقع إنتاجها ، فقد بدأت ، وما زالت مستمرة حتى وقتنا الراهن ، الأبحاث تتجه في اتجاهين في محاولة محاكاة العقل البشري :

الاتجاه الأول : يحاول تمثيل الحاسوب كشبكات عصبية وهو ما يعرف
Artificial Neural Network .

الاتجاه الثاني : فيحاول من خلال التعاون مع علماء الهندسة الوراثية في إنتاج رقيقة حيوية وذلك بتكييف البروتينات لتحل محل السيليكون في

الدوائر الإلكترونية ، وتخدم أبحاث هذين الاتجاهين حواسيب الجيل السادس المتوقع إنتاجها .

أضف إلى ما سبق تطور أساليب الاتصالات وشبكات الحواسيب ونخص بالذكر شبكات الأنترنت Internet Network والتي سنتناولها في فصول لاحقة، والسؤال الذي يبقى مطروحاً : كم من هذه الأبحاث والمشاريع والمبادرات سيكتب لها النجاح ؟؟.

والجواب : من الصعب التنبؤ بالمستقبل ولكن من المؤكد أن تطور شبكات الأنترنت وشبكات الاتصالات الحاسوبية الأخرى ستلعب دوراً حاسماً في هذا الصدد، وكذلك التقدم المتزايد في تكنولوجيا العتاد والبرمجيات ، ومن المؤكد أيضاً أن السنوات القادمة من عمر الحواسيب الشخصية لن تكون أقل إثارة من تلك السنوات التي مضت ، وأن غداً لناظره قريب !

تمارين وأسئلة عامة

1. عدد مع الشرح العناصر الأساسية للنظام الحاسوبي ؟
2. عرف الحاسوب وبين علاقته بالإنسان ؟
3. تكلم بإيجاز عن أهم ميزات وإمكانيات الحاسوب ؟
4. هل يمكن للحاسوب أن يفكر بدلاً عن الإنسان ، ولماذا ؟
5. بين الفرق بين الحواسيب الرقمية والحواسيب القياسية ؟
6. اذكر أهم ميزات الحواسيب الميني ؟
7. عدد خمساً من استخدامات الحاسوب في الحياة ؟
8. اختر الجواب الصحيح من الأسئلة التالية :
 - 1- وظيفة الحاسوب هي :
 - أ- معالجة البيانات .
 - ب- تخزين البيانات .
 - ج- إخراج المعلومات .
 - د - كل ما ذكر .
 - 2- ظهرت الأقراص المرنة كوسيلة للتخزين في :
 - أ- الجيل الأول .
 - ب- الجيل الثاني .
 - ج- الجيل الثالث .
 - د- الجيل الرابع .
 3. استخدمت لغة للتجميع في برمجة الحواسيب بشكل واسع في :
 - أ- الجيل الأول .
 - ب- الجيل الثاني .
 - ج- الجيل الثالث .
 - د- الجيل الرابع .
 - 4- يطلق اسم حواسيب الدارات المتكاملة على حواسيب الجيل :
 - أ - الأول .
 - ب - الثاني .
 - ج - الثالث .
 - د - الرابع .

5- ظهرت المعالجات الميكروية Micro Processor مصاحبة للحواسيب:

أ- المتوسطة الحجم . ب- المصغرة .

ج- الكبيرة . د- العملاقة .

6- الأساس التكنولوجي للحاسوب العصري هو :

أ- الترانزستور . ب- الدارات المتكاملة .

ج- الصمامات المفرغة . د- المفتاح .

7- ظهرت الحواسيب المصغرة (Micro computers) مصاحبة لتكنولوجيا :

أ- الصمامات المفرغة . ب- الدوائر المتكاملة .

ج- الترانزستور . د- أ + ب .

8. أستخدمت حواسيب الجيل الثاني :

أ- ذاكرة الحلقات المغناطيسية . ب- ذاكرة شبه موصل .

ج- ترانزستور . د- الدارات المتكاملة .

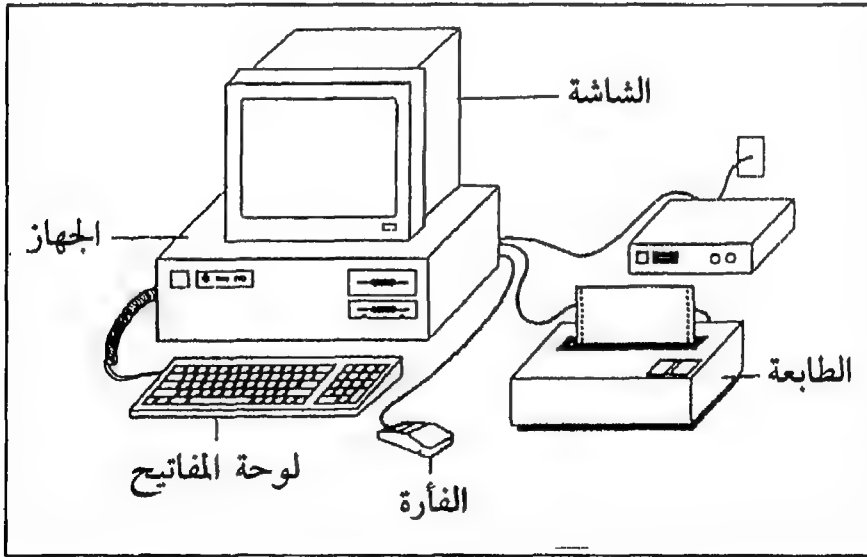
الفصل الثاني

مكونات نظام الحاسوب

Components of Computer System

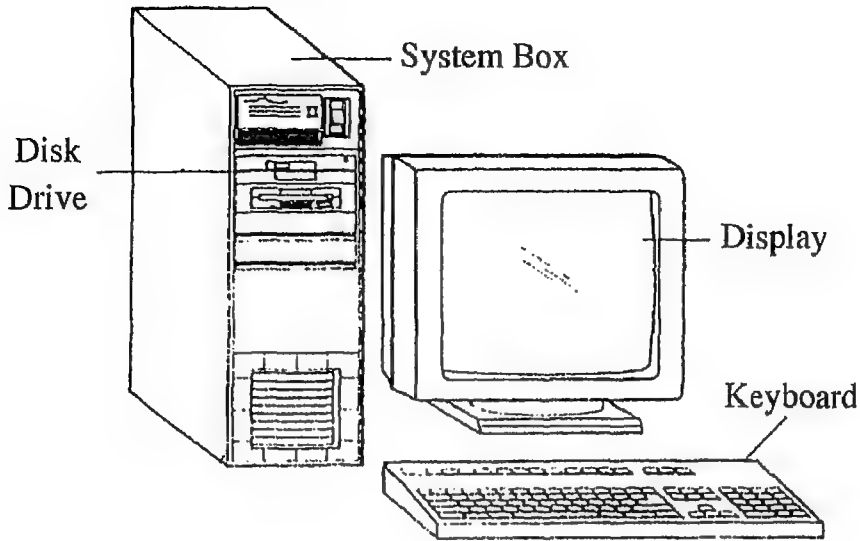
(1 - 2) مقدمة

يتكون أي حاسوب من العديد من الأجزاء التي تعمل سوية لإنجاز العديد من المهام المختلفة ، كما في الشكل (1) ، سوف نقوم بشرح المكونات و آلية عملها بشكل مفصل.



الشكل (1) المكونات الأساسية لجهاز الحاسوب

هناك نماذج أخرى للحاسبات الشخصية الشائعة الاستعمال منها النموذج
الموجود في الشكل (2) :

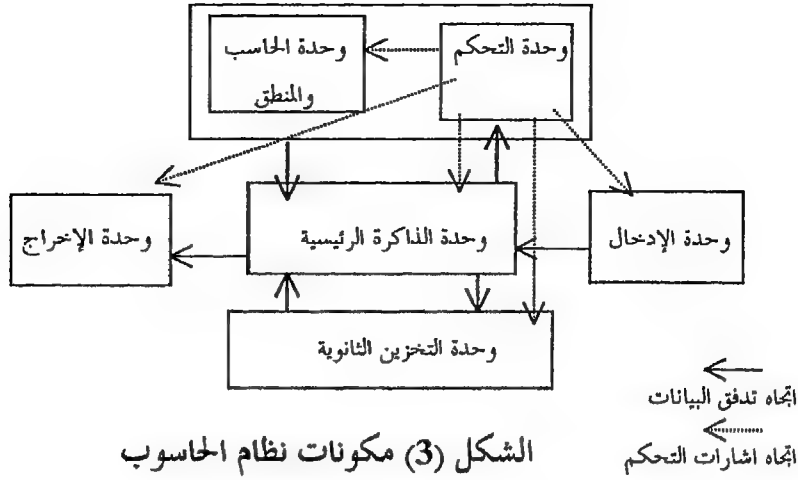


الشكل (2) أحد نماذج الحاسبات الشخصية الشائعة

إن أي حاسوب يتكون من الأجزاء الأساسية التالية كما هو واضح في
الشكل (3) :

- | | |
|-------------------------------|----------------------------|
| Central Processing Unit (CPU) | 1 - وحدة التشغيل المركزية |
| Main Memory | 2 - وحدة الذاكرة الرئيسية |
| Secondary Storage | 3 - وحدات التخزين الثانوية |
| Input Units | 4 - وحدات الإدخال |
| Output Unit | 5 - وحدات الإخراج |

وحدة المعالجة المركزية



(2 - 2) وحدة المعالجة المركزية

Central Processing Unit (CPU)

يطلق على وحدة الحساب و المنطق و وحدة التحكم معاً إسم وحدة المعالجة المركزية أو المعالج Processor، والمعالج في الحاسوب كالدماع في الانسان فهو الذي يفكر ويحلل المعلومات ويتخذ القرارات ، إلى ما هنالك من أعمال أخرى . للمعالج عدة أنواع ، أشهرها معالجات الحواسيب الشخصية من انتاج شركة INTEL ، وسنبين فيما يلي أهم المعالجات و سرعتها وفي أي الحواسيب الشخصية استخدمت :

نوع المعالج CPU	سرعة المعالج Speed	حجم ذاكرة المعالج Memory	نوع الحاسوب Computer
80.8085	0.4 MHZ	256 KB	Pc
8086.8088	4-10 MHZ	640 KB	XT
80286	12-20 MHZ	1 MB	AT
80486	16-40 MHZ	1 MB	AT 386
80486	23-100 MHZ	4 MB	AT 486
80586 (Pentium)	60-400 MHZ	8,16,32,64, 128,256 MB	Pentium

من الجدير بالذكر أن للمعالج Pentium عدة أنواع حسب السرعة ،
منها : بنتيوم 75 ، بنتيوم 100 ، بنتيوم 133 ، بنتيوم 166 ، بنتيوم 200 ، بنتيوم
266 ، وبنتيوم 300 ، وبنتيوم 400 وبنتيوم 450 .

تعتبر وحدة المعالجة المركزية من الوحدات الأساسية في الحاسوب فهي
تمثل القلب النابض للحاسوب لأنها تحتوي على جميع الامكانيات الضرورية لتداول
البيانات والتعليمات و ضبط جميع العمليات من حساب وتخزين و استرجاع . كما
تعتبر من الوحدات الأساسية التي تحدد قدرة و سرعة و ثمن الحاسوب .
(2 - 2 - 1) وحدة الحساب والمنطق

Arithmetical and Logical Unit

تتكون وحدة الحساب و المنطق من :

- أ. مجموعة من الدوائر الالكترونية المنطقية التي يتم توظيفها لأداء العمليات
التي تتم داخل ذاكرة الحاسوب .
- ب. مجموعة من المسجلات Registers، تستخدم هذه الوحدة لاستيعاب
البيانات التي يجري معالجتها ، و كذلك استيعاب النتائج الوسيطة اثناء
عملية المعالجة ، ويمكن التمييز بين أنواع المسجلات التالية التي تضمها
وحدة الحساب و المنطق :

1. مسجلات النقطة العائمة Floating Point Registers والتي
تستخدم لإنجاز العمليات الحسابية على الارقام الحسابية ذات النقطة
العائمة والتي سيتم شرحها لاحقاً .
2. مسجلات النقطة الثابتة Fixed Point Registers و التي تستخدم
لإنجاز العمليات الحسابية على الارقام الحسابية ذات النقطة الثابتة .

3. المسجل المراكم Accumulator وهو مسجل خاص يستخدم لتجميع نتائج العمليات الحسابية التي تنجزها هذه الوحدة ويعتبر المراكم هو المسجل الرئيسي في وحدة الحساب والمنطق ، ويدعى أيضاً المسجل .

4. مسجلات الإزاحة Shifting Registers تستخدم في إنجاز عمليات الإزاحة .

5. عدد من المسجلات عامة الغرض General Purpose Registers تساعد في إنجاز المهام المنوطة بهذه الوحدة ، و عادة ما تعنون هذه المسجلات بعنوان رقمي يبدأ بالرقم 0,1,2,3,4,... .
وظيفة وحدة الحساب والمنطق :

- وحدة الحساب والمنطق مسؤولة عن إنجاز ما يلي :
- كافة العمليات الحسابية مثل الجمع والطرح والقسمة والضرب والرفع للأس ..
- كافة العمليات المنطقية الأساسية OR , AND , NOT .
- عمليات المقارنة وتشمل أكبر من ، و أصغر من ، أصغر أو يساوي ، أكبر أو يساوي ، يساوي ، لا يساوي .
- عمليات الإزاحة SHIFT والمقصود بها تحريك محتويات الوحدة التخزينية إلى اليمين أو اليسار لأغراض المعالجة مع الأخذ بعين الاعتبار إن الإزاحة نحاة واحدة إلى اليسار تعني ضرب الرقم ب 2 و الإزاحة إلى اليمين تعني القسمة على 2 (وهو أساس النظام الثنائي) .

مثال يوضح آلية عمل وحدة الحساب والمنطق :
لتكن لدينا المعادلة التالية :

$$Z = X + Y$$

ولنرى كيف يتم إنجاز عملية الحساب في هذه المعادلة .

يتم تخزين قيمة X وقيمة Y أولاً في الذاكرة ومن ثم يتم إنجاز عملية الجمع وفقاً للتتابع التالي:

- 1- يتم نقل قيمة X إلى وحدة الحساب والمنطق حيث يتم تخزينها في المسجل A.
- 2- تقوم الدوائر المنطقية بجمع قيمة Y وإضافتها إلى قيمة المراكم .
- 3- يتم تخزين ناتج الجمع في المراكم .
- 4- يتم نقل محتويات المراكم إلى المخزن Y في الذاكرة .

واضح من التتابع السابق أن المراكم نفسه لا يقوم بالعملية الحسابية ولكنه يستخدم في تجميع نتائج العمليات تمهيداً لتخزينها في الذاكرة ، يكون طول المراكم مساوياً لطول الوعاء التخزيني في الذاكرة ، لذلك فإنه في العمليات الحسابية التي يحتاج نتائجها الوسيطة إلى وعاء تخزيني أكبر مثل عمليات الضرب يتم الحاق مسجل اضافي بالمراكم ليصبح في هذه الحالة مراكماً مضاعف السعة .

(2-2-2) وحدة التحكم : CONTROL UNIT

إذا كانت وحدة المعالجة المركزية هي قلب الحاسوب فإن وحدة التحكم هي قلب وحدة المعالجة المركزية فهي التي تقوم بعملية التنسيق بين أعمال وحدات الحاسوب الأخرى بغرض تنفيذ البرنامج المخزن مسبقاً في الذاكرة وبالتالي فإن دورها يبدأ في بداية التشغيل ولا ينتهي إلا بانتهائه .

مكونات وحدة التحكم :

تتكون وحدة التحكم من العناصر التالية :

- 1- عناصر توقيت الأعمال التي يؤديها الحاسوب وتقوم بالمهام التالية :
 - ضبط التوقيت لتسلسل تعليمات البرنامج .
 - نقل التعليمات التالية في التنفيذ من موقعه في الذاكرة إلى وحدة المعالجة المركزية.
 - تحليل التعليمات وتحديد العملية المراد تنفيذها .

- اصدار أوامر للتحكم إلى باقي وحدات الحاسوب المعنية في التنفيذ على شكل اشارات تحكم وتطلب منها اداء العملية المراد تنفيذها .

2-عناصر متابعة وتنفيذ الأوامر الحاسوبية وهي عبارة عن مجموعة الدارات الالكترونية التي تتحد مع بعضها لتكون ما يسمى بالمسجلات ، أهم هذه المسجلات :

- مسجل الأمر الحالي Instruction Register وباختصار (IR) وهو الذي يستخدم في تخزين التعليمة التي يجري تنفيذها بعد جلبها من الذاكرة

- عداد البرنامج Program Counter وباختصار (PC) وهو عبارة عن مسجل يحتفظ بعنوان التعليمة التالية في التنفيذ ، ويقوم بتعداد الأوامر التي يجري تنفيذها من خلال تحديده للأمر التالي في التنفيذ، ولتوضيح ذلك نفترض اننا بصدد تشغيل برنامج موجود فعلا في الذاكرة وكانت محتويات العداد صفرا ، فان ذلك يعني ان الأمر التالي في التنفيذ هو الأمر الذي عنوانه صفر ، يزداد العداد بالقيمة 1 فيصبح الأمر التالي لتنفيذ هو 1...الخ

- محلل التعليمة Instruction Decoder باختصار (ID) وهو عبارة عن مسجل يعمل على تحليل شيفرة التعليمة بهدف تحديد نوع العملية المراد تنفيذها وهذا المسجل هو عبارة عن دائرة إلكترونية لها مدخل وعدد من المخارج مساو لعدد العمليات التي يمكن لوحدة المعالجة المركزية القيام بتنفيذها.

وظائف وحدة التحكم :

إن وحدة التحكم كما قلنا هي التي تقوم بالتنسيق بين وحدات الحاسوب الأخرى و ضبط كافة العمليات التي تتم داخل وحدة المعالجة المركزية ويمكن بالإضافة إلى ذلك تحديد مهام هذه الوحدة في الآتي :

1. يتم ادخال البيانات و التعليمات إلى ذاكرة الحاسوب بناءً على إشارات

خاصة من وحدة التحكم .

2. تقوم وحدة التحكم بجلب الأمر التالي في التنفيذ من موقعه في الذاكرة و إعادة تخزينه في المسجل المعد لهذا الغرض .

3. اصدار إشارات إلى كل من وحدة الحاسب و المنطق وكذلك الذاكرة من أجل البدء بتنفيذ الأمر ، وإعادة تخزين النتائج في الذاكرة تمهيداً لإخراجها على إحدى وسائط الإخراج .

4. تحديد عنوان الأمر التالي في التنفيذ .

5. يتم تكرار هذه العمليات حتى ينتهي البرنامج .

دورة المعالجة في وحدة التحكم :

Processing Cycle in control unit :

يتم تنفيذ التعليمات والأوامر داخل الحاسوب في دورتين رئيسيتين لوحدة التحكم هما :

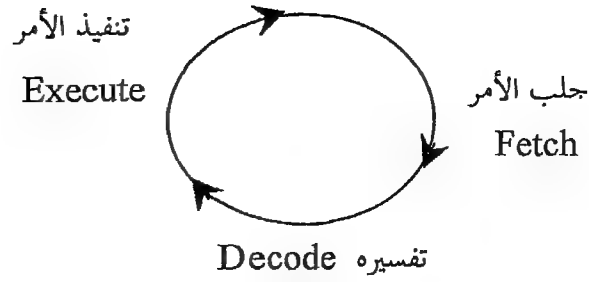
أ. دورة التعليمة Instruction Cycle : أي تحديد الأمر و التعليمة المراد معالجتها ، ويمكن إيجازها كما يلي :

1. جلب Fetch الأمر أو التعليمة من الذاكرة و تخزين عنوانها في المسجل (PC) وتخزينها في المسجل (IR) .

2. تفسير التعليمة و فك شيفرتها Decoding و تحديد نوع العملية المراد تنفيذها .

3. اصدار الاشارات اللازمة لتنفيذ هذه التعليمة و لمعرفة التعليمة التالية في التنفيذ.

4. تكرار الخطوات السابقة حتى تنتهي الأوامر و التعليمات ، الشكل (4) يلخص هذه الدورة .



الشكل (4)

ب. دورة التنفيذ Execution Cycle :

حيث يتم في هذه الدورة جلب البيانات طبقاً للأوامر أو التعليمات بواسطة وحدة التحكم، إلى وحدة الحساب والمنطق حيث تتم معالجتها ومن ثم إعادة تخزين النتائج المرحلية في مسجلات معينة وإعادة تخزين النتائج النهائية في ذاكرة الحاسوب الرئيسية.

أولويات تنفيذ العمليات الحسابية داخل وحدة المعالجة الرئيسية:

تقوم وحدة الحساب والمنطق بانجاز العمليات على التعابير الحسابية وفق ترتيب محدد من قبل البرمجيات ، و يتم تنفيذه تحت اشراف وحدة التحكم على الشكل التالي :

1. الرفع الى أس .
2. الضرب و القسمة .
3. الجمع و الطرح .

إذا استخدمت الأقواس في التعابير الحسابية فان ما بين القوسين يتم تنفيذه أولاً وحسب قاعدة الأولويات ، وفي حال وجود أقواس متداخلة يتم تنفيذ القوس الداخلي أولاً ثم الخارجي .

مثال 1 : يتم تنفيذ التعبير الحسابي التالي :

$$A - B + 8 / 3 * C ** 4$$

كما يلي :

أ- يتم تنفيذ $C ** 4$ أولاً .

ب- في الخطوة الثانية يتم تنفيذ $8 / 3$.

ت- في الخطوة الثالثة يتم تنفيذ $8 / 3 * C ** 4$ أي ضرب

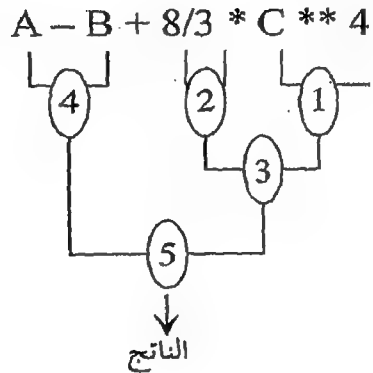
ناتج (أ) بناتج (ب).

ث- في الخطوة الرابعة يتم تنفيذ $A - B$.

ج- في الخطوة الخامسة يتم تنفيذ

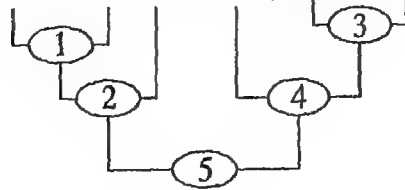
$$A - B + 8 / 3 * C ** 4$$

أي جمع ناتج (ث) مع ناتج (ت) .



مثال 2 : يتم تنفيذ التعبير الحسابي التالي :

$$(A - B + 18) / (15 * C ** 6)$$



كما يلي :

أ. يتم تنفيذ $A - B$ داخل القوس الأول .

ب. يتم تنفيذ العملية الحسابية داخل القوس الأول

$$A - B + 18$$

ت. يتم تنفيذ عملية الرفع إلى اسس داخل القوس الثاني

$$C ** 6$$

ث. يتم تنفيذ العملية الحسابية داخل القوس الثاني

$$15 * C ** 6$$

ج. الخطوة الاخيرة يتم تنفيذ عملية القسمة .

$$(A - B + 18) / (15 * C ** 6)$$

(2 - 3) وحدات التخزين الداخلية Internal storage Units

تتعدد وحدات التخزين الداخلية وتتنوع في الشكل والسعة والوظيفة ،

ويمكن التمييز بين الأنواع التالية :

1- الذاكرة المؤقتة Random Access Memory (RAM) .

2- الذاكرة الدائمة Read Only Memory (ROM) .

3- الذاكرة الفورية Cache Memory .

4- المسجلات Registers .

5- المخازن الانتقالية Buffers .

(2-3-1) الذاكرة الرئيسية Main Memory

وهي ذاكرة القراءة والكتابة ، حيث يشار اليها بمسميات متعددة منها :

أ - المخزن الاساس Primary Storage .

ب - المخزن الرئيسي Main Storage .

ج- ذاكرة الوصول العشوائي Random Access Memory .

عناصر بناء الذاكرة الرئيسية :

هناك العديد من العناصر التي استخدمت لبناء الذاكرة الرئيسية للحاسوب، فقد تم استخدام الصمامات في أول حاسوب ، كما تم استخدام الاسطوانات المغنطة في منتصف الخمسينات ، كما استخدمت الحلقات المغنطة خلال الفترة الواقعة في بداية الستينات إلى منتصف السبعينات منذ منتصف السبعينات بدء استخدام أنصاف النواقل لبناء الذاكرة .

إن الذاكرة المصنوعة من أنصاف النواقل تفقد كل ما هو مخزن بها عند فصل التيار الكهربائي عن الحاسوب ، ولهذا السبب فإن أنظمة الحاسبات الكبيرة التي تستخدم هذا النوع من الذاكر تستخدم نظام تغذية احتياطي Uninterruptible Power Supply واختصاراً (UPS) وذلك من أجل الحفاظ على محتويات الذاكرة الرئيسية لفترة زمنية أطول في حالة الانقطاع المفاجئ للتيار الكهربائي .

الذاكرة الرئيسية هي ذاكرة القراءة والكتابة :

تتكون الذاكرة الرئيسية من عدد من المواقع التخزينية ، لو فرضنا أن عددها N فإنها تكون مرقمة من 0 إلى $N - 1$ ، يدل على عنوان الموقع ، يستخدم العنوان للوصول إلى الموقع التخزيني ، وعندما يتم تخزين أي عنصر جديد من البيانات أو التعليمات في موقع تخزين فإن المحتويات السابقة في هذا الموقع يتم مسحها وإزالتها وحفظ ما هو جديد بدلاً منها ، ويطلق على عملية ادخال بيانات أو تعليمات جديدة إلى الذاكرة عملية الكتابة (Writing) ، وعندما يتم استرجاع البيانات والتعليمات من موقع التخزين بالذاكرة الرئيسية فإنه يتم قراءتها (Reading) ثم عمل نسخة منها في مكان آخر ولا يتم إزالتها من موقع التخزين الرئيسي .

يمكن تشبيه عملية القراءة بقراءة الإنسان لكتاب ما ، فهو يقرأ منه وقت ما يشاء ولكنه لا يغير موقع الكتابة فيه ، أما عملية الكتابة فيمكن تشبيهها بعملية كتابة الإنسان في الكراسة حيث يستطيع الكتابة فيها ثم يقرأ منها ما كتبه كما يمكننا مسح أي جزء ثم كتابته وإضافة جزء جديد بدلاً منه ونظراً لإمكانية الكتابة على الذاكرة الرئيسية وإمكانية القراءة منها فإنه يطلق عليها أسم ذاكرة القراءة والكتابة .

كل موقع في الذاكرة يتكون من عدد متساوٍ من الخلايا يخصص كل منها لحفظ الرقم "0" أو الرقم "1" ، فتسمى الخلية الواحدة بثنائية bit (وهي أصغر وحدة تخزين تقاس بها سعة الحاسوب) ، لذلك فإن حجم الموقع يحدد بعدد الثنائيات التي يتكون منها ، وهذا يختلف من حاسوب لآخر .

عادة يتكون الموقع التخزيني المرتبط بعنوان من 8 ثنائيات (أي bit 8) والذي يدعى نظام ترميز معين نذكر منها شيفرة الاسكي ASCII وشيفرة اليبسك EBCDIC .

ملاحظة هامة

- أن وحدة التخزين هي الثنائية Bit .

- لتمثيل رمز وحرف في الحاسوب يستخدم ما يسمى البايت Byte

$$1 \text{ byte} = 8 \text{ bit}$$

- لتخزين عدد او جمل ذات معنى تستخدم ما يسمى Word وحيث ان حجم

ال Word يتراوح من 8 ثنائيات إلى 64 ثنائية ففي الحواسيب الصغيرة

يكون حجم الكلمة 8 ثنائيات أي Byte

و في الحواسيب الأكبر يكون حجم الكلمة يساوي 16Bit ويساوي 2Byte

و في الحواسيب الأكبر يكون حجم الكلمة يساوي 32Bit ويساوي 4Byte

و في الحواسيب الأكبر يكون حجم الكلمة يساوي 64Bit ويساوي 8Byte

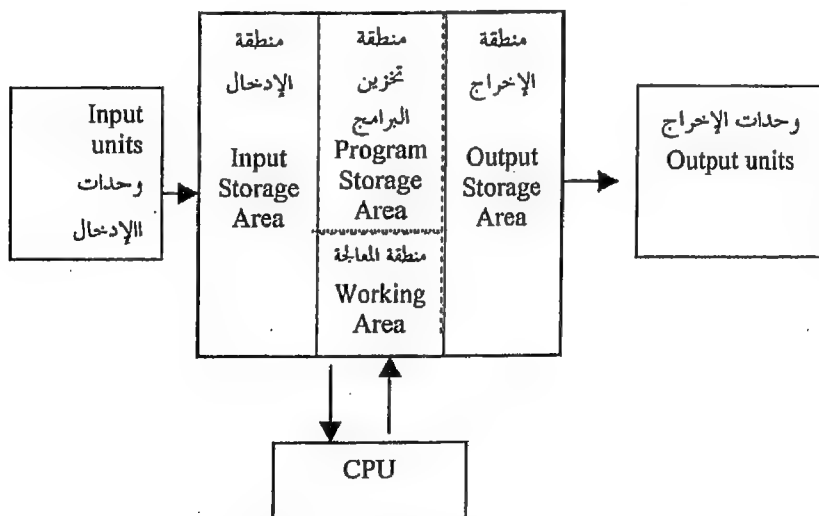
ولهذا فإن Word تتكون من 4Byte وكل 1Byte = 8 Bits

- يعتبر حجم الكلمة إحدى الصفات الرئيسية للحاسوب ، حيث ترتبط وحدة الذاكرة الرئيسية بوحدة المعالجة المركزية بواسطة ناقلات Buss ، يتم من خلالها تبادل المعلومات ، فكلما كانت حجم الكلمة أكبر كلما كانت عملية نقل البيانات بين وحدة الذاكرة ووحدة المعالجة المركزية تتم بشكل أسرع .

- الناقل Buss عبارة عن أسلاك تربط وحدة المعالجة المركزية ووحدة الذاكرة وكذلك لربط الذاكرة مع أجهزة الإدخال و الإخراج ووحدة التخزين الثانوية.

مناطق الذاكرة :

تنقسم الذاكرة الرئيسية إلى مناطق ، كما هو موضح بالشكل (5) حيث لا تفصل هذه المناطق أي فواصل أو حدود مادية ، فالخطوط المتقطعة بالشكل (5) ليست فواصل طبيعية ولكنها خطوط وهمية ، إذ أن حجم كل منطقة من هذه المناطق يختلف من برنامج إلى آخر و هذه المناطق هي :



الشكل (5) يمثل مناطق تقسيم الذاكرة الرئيسية

1- منطقة البرامج Program Area :

وهو مخصص لتخزين البرامج والأوامر والتعليمات سواء تلك المتعلقة بنظام التشغيل أو البرامج التطبيقية .

2- منطقة المدخلات Input Area :

مخصص لاستقبال البيانات من وحدات الإدخال وتسمى هذه المنطقة Input buffer حيث Buffer هنا هي وحدات تخزينية مخصصة لاستقبال البيانات النشطة وتخزينها مؤقتاً إلى أن يتم معالجتها .

3- منطقة المخرجات Output area :

هذه المنطقة مخصصة لاستقبال النتائج بعد معالجتها وتمهيداً لإخراجها وتسمى هذه المنطقة أيضاً Output Buffer إذا أن التخزين فيها يكون تخزيناً مؤقتاً للمعلومات النشطة تمهيداً لإخراجها بواسطة وحدة الإخراج المناسبة .

4- منطقة العمل (المعالجة) Working area :

وتعمل هذه المنطقة بمثابة هامش أو مسودة حيث يتم فيها تخزين النتائج الوسيطة والبيانات المفترضة التي لا تعرف ضمن المدخلات .

معايير التمييز بين الذاكرات :

من أهم العوامل الأساسية التي يمكن ان تتميز بها الذاكرة ما يلي :

1- السعة Capacity :

ويقصد بها عدد المواقع التخزينية في الذاكرة ، و معظم الحواسيب الصغيرة تقاس

سعة ذاكرتها بالبايت أي : $1 \text{ Byte} = 8 \text{ Bit}$.

و في بعض الاخر تقاس بالكيلوبايت Kilo Byte حيث :

$$1 \text{ K.B} = 2^{10} \text{ Byte} = 1024 \text{ Byte}$$

و بعض الأجهزة تقاس بالميجابايت Mega Byte حيث :

$$1 \text{ M.B} = 2^{10} \text{ K.B} = 2^{20} \text{ Byte}$$

واحيانا تقاس بالجيجا بايت Giga Byte حيث :

$$1 \text{ G.B} = 2^{10} \text{ M.B} = 2^{20} \text{ K.B} = 2^{30} \text{ Byte}$$

وفي الوقت الراهن أصبح يستخدم مقياس تيرا بايت Tera Byte حيث :

$$1 \text{ T.B} = 2^{10} \text{ G.B} = 2^{40} \text{ Byte}$$

2- زمن الوصول Access Time :

وهو الزمن الذي يستغرقه الحاسوب في قراءة كلمة واحدة (Word) من الذاكرة ، كلما زادت سرعة الوصول زادت سرعة الحاسوب في تنفيذ تعليمات البرنامج المعالج وبالتالي زاد مستوى أداء الحاسوب .

3- طريقة الوصول Access Method :

يقصد بها كيفية الوصول إلى المعلومات المخزنة ويتم ذلك بطريقتين :

أ- طريقة الوصول المباشر أو العشوائي Random Access Method : أي الوصول إلى المعلومات المطلوبة مباشرة دون الحاجة إلى المرور على ما يسبقها من بيانات .

ب- طريقة الوصول غير المباشر أو التتابعي Sequential Access Method : أي الوصول إلى المعلومات المطلوبة بشكل تسلسلي (تتابعي) بعد المرور على معلومات جميع المواقع السابقة للموقع المطلوب .

(2-3-2) الذاكرة الفورية Cache Memory

وتسمى أيضا بالذاكرة المسرعة وتستخدم خلال عملية التشغيل وهي عبارة عن ذاكرة تخزين مؤقت ذات سرعة عالية جداً تفوق سرعة الذاكرة RAM بكثير. وهي تستخدم للتخزين المؤقت للبيانات والتعليمات المطلوب استرجاعها مرات عديدة أثناء عملية معالجة البيانات مما يساعد على سرعة معالجة البيانات

(2-3-3) الذاكرة الدائمة Read Only Memory ROM

وتسمى بذاكرة القراءة فقط ، حيث أن استخدامها يقتصر على القراءة فقط ومن مهامها الرئيسية الإشراف على بدء تشغيل الحاسوب . من خصائص هذه الذاكرة :

1. تحتوي على البرامج والروتينات التي تحتاج إليها بصفة دائمة ، مثل :
 - مترجم (مفسر) اللغة الرئيسية للجهاز كمفسر لغة بيسك في معظم الأجهزة المصغرة .

- روتينات الإدخال و الإخراج (Input and Output Routines)

- روتينات بدء تشغيل الجهاز (Start-up Routines)

2. يمكن القراءة من هذه الذاكرة فقط وذلك عند الحاجة لأي من البرامج أو الروتينات السابقة ، ولا يمكن لمستخدم الجهاز أن يسجل في الذاكرة أية معلومات (يقرأ منها ولا يكتب فيها) .

3. لا تفقد محتوياتها سواء بعد القراءة منها أو عند فصل التيار الكهربائي .

4. محتويات هذه الذاكرة لا تتغير مطلقا من قبل المستخدمين لها .

أهم أشكال ذاكرة ROM :

1. ذاكرة ROM : هي ذاكرة مبرمجة لمرة واحدة ، أي يخزن فيها برنامج ثابت أو معلومات ثابتة ولا يمكن اجراء أي تعديل على مضمونها .

2. ذاكرة القراءة القابلة للبرمجة

Programmable Read Only Memory (PROM)

بمجرد برمجة هذه الذاكرة فإنه لا يمكن تغيير ما هو مخزن بها ، أي أن الذاكرة

PROM تصبح ROM وما هو مخزن عليها يمكن قراءته فقط .

3. ذاكرة القراءة القابلة للبرمجة و المسح (EPROM)

Erasable and Programmable Read Only Memory

كما هو واضح من التسمية فان ما تحمله هذه الذاكرة يمكن مسحه وإعادة برمجته من جديد لتسجيل تعليمات جديدة باستخدام وسائل خاصة للبرمجة وتتم عملية المسح بتعريض شريحة EPROM للأشعة فوق البنفسجية ، وهذه الذاكرة مثل الذاكرة ROM يمكن قراءة ما هو مسجل فيها فقط وتبقى هذه الذاكرة إلى أن يتم مسحها بالطريقة التي سبق ذكرها .

(2-3-4) المسجلات Registers :

تطلق تسمية مسجلات على الوحدات التخزينية الموجودة في وحدتي التحكم والحساب والمنطق ، ويمكن تعريف المسجل على انه : وحدة إلكترونية قادرة على استيعاب وحدة من المعلومات أثناء عملية المعالجة .

تعتبر المسجلات وحدات تخزينية مؤقتة . بمعنى أن المعلومات تخزن فيها لفترة وجيزة حتى تتم معالجتها ، كما أنها تتميز بسرعتها الفائقة في نقل المعلومات وتداولها بين وحدة الحساب والمنطق ووحدة التحكم والذاكرة .

عادة يكون طول المسجل مساويا لطول الكلمة Word في الذاكرة ، وهذا منطقي إذ أن المسجل يستخدم للتخزين المؤقت للبيانات والتعليمات التي تنتقل اليه من الذاكرة أو تنتقل منه إليها . ويختلف عدد المسجلات وأطوالها وأنواعها من نظام إلى آخر وفيما يلي بعض أنواع المسجلات ووظيفة كل منها:

1- المراكم Accumulator : وقد يسمى A - Register ، يستخدم في تجميع النتائج الحسابية الوسيطة ، وهو الرئيسي في وحدة الحساب والمنطق.

2- مسجل العنوان Address Register ، وهو يستخدم لتخزين عنوان أو رقم الموقع التخزيني بالذاكرة الذي يراد التعامل معه ، Memory Address Register ومسجل محتوى الذاكرة .

3- مسجلات وحدة التحكم : مثل مسجل التعليمات Instruction Register ومسجل عداد البرامج Program Counter Register ومسجل تحليل التعليمات Instruction Decoder .

4- مسجلات عامة الغرض General Purpose Register : وتستخدم لتخزين أنواع مختلفة من البيانات .

5- مسجلات القراءة و الكتابة في الذاكرة RAM وهذه المسجلات هي : أ. مسجل محتويات الذاكرة : Memory Buffer Register (MBR)

ب.مسجل عنوان الذاكرة : Memory Address Register (MAR)

(2 - 3 - 5) المخازن الانتقالية Buffers :

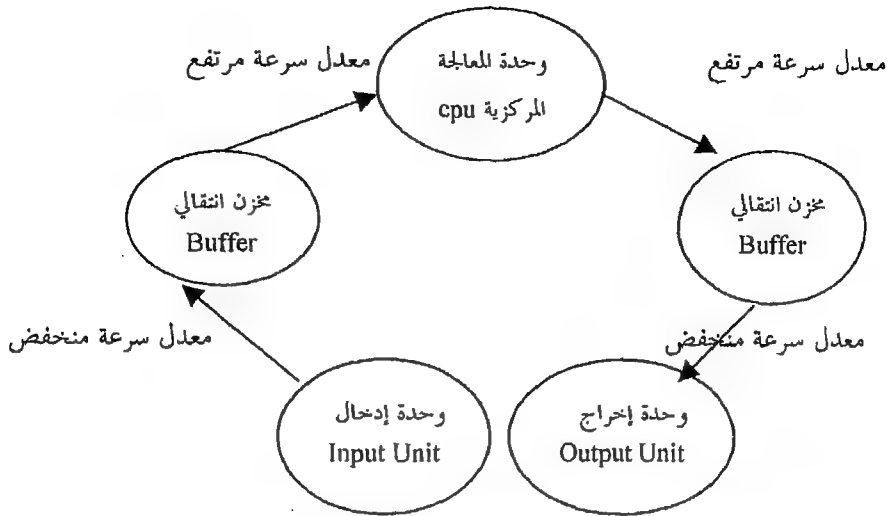
المخزن الانتقالي هو مساحة تخزينية بالذاكرة أو خارجها ذات سعة محدودة وسرعة تداوله عالية ، تستخدم لاستيعاب البيانات استيعاباً مؤقتاً تمهيداً لنقلها وتداولها بين أي وحدتين من وحدات الحاسوب . وتسمى هذه العملية بعملية التخزين الانتقالي طبقاً لهذا المفهوم فإن المسجلات بوحدة المعالجة المركزية تعتبر بصورة أو بأخرى مخازن انتقالية ، ذلك لأنها تستخدم في الاحتفاظ بالبيانات والتعليمات احتفاظاً مؤقتاً تمهيداً لتداولها بين وحدات التحكم ، الذاكرة ، الحساب والمنطق إلا أنه جرى العرف الاصطلاحي على تسميتها مسجل ، أما اصطلاح Buffers فإنه يطلق على النوعين التاليين من المخازن الانتقالية :

أ- المخازن الانتقالية الملحقه بوحداث الإدخال والإخراج :

تستخدم هذه المخازن لتقليل من اثر فارق السرعة بين وحدات الإدخال والإخراج من ناحية ووحدة المعالجة المركزية من ناحية أخرى . فمثلاً عند إدخال بيانات عن طريق لوحة المفاتيح فإنه مهما بلغت سرعة مدخل البيانات فإنها لا تتعدى حرفاً أو حرفين في الثانية الواحدة وهذه السرعة أقل بكثير من سرعة وحدة المعالجة المركزية ، حيث يتم استقبال البيانات وتجميعها في كتل من البيانات (سطور مثلاً) ومن ثم نقلها من هذا المخزن الانتقالي إلى وحدة المعالجة المركزية كتلة بكتلة وبسرعة عالية تتناسب وسرعة وحدة المعالجة المركزية .

كذلك الحال عند إخراج المعلومات من وحدة المعالجة المركزية ، إذ أن سرعة وحدة الطباعة مثلاً تكون بطيئة جداً بالنسبة لسرعة المعالج المركزي مما يؤدي إلى تعطيل وقت المعالج نتيجة لتأخر وحدة الطباعة وعدم قدرتها على مجاراة المعالج في سرعة نقل البيانات وإخراجها . لذلك يستخدم مخزن إنتقالي يقع بين وحدة الإخراج ووحدة المعالجة المركزية ، حيث تنقل البيانات إليه بسرعة عالية هي سرعة

التداول بوحدة المعالجة المركزية التي تتفرغ لتنفيذ عملية أخرى ويتولى المخزن الانتقال نقل النتائج إلى وحدة الطباعة حسب سرعتها والشكل (6) يبين ذلك .



الشكل (6) يمثل المخازن الانتقالية الملحقة بوحدات الإدخال والإخراج
ب- المخازن الانتقالية بالذاكرة الرئيسية :

عند نقل البيانات من وحدة التخزين الخارجية مثل الأقراص والأشرطة الممغنطة إلى وحدة المعالجة المركزية أو العكس ، تنشأ أيضاً مشكلة فارق السرعة بينها وللتغلب على هذه المشكلة يتم تخصيص قطاعات بالذاكرة الرئيسية تستخدم كمخازن انتقالية .

فالبيانات في الملف تكون منظمة في وحدات بيانات تسمى سجلات Records ، ويتم عادة تجميع عدد من السجلات لتعامل كوحدة واحدة تسمى كتلة .

عند ادخال البيانات إلى الحاسوب من الملف يتم نقل كتلة كاملة من البيانات إلى المخزن الانتقالي بالذاكرة الذي يسمى في هذه الحالة مخزن المدخلات الانتقالي (Input Buffer) ، حيث تقوم وحدة المعالجة المركزية بالتعامل مع محتويات هذا المخزن الانتقالي سجلاً بعد سجل .

كذلك عند إخراج المعلومات من الحاسوب إلى الملف ، فإنه يتم تجميع سجلات المعلومات على شكل كتل بمخزن انتقالي بالذاكرة الذي يسمى في هذه الحالة مخزن المخرجات الانتقالي (Output Buffer) ، ومن ثم يتم نقل محتوياته إلى الملف بوحدة التخزين الخارجي .

(2 - 4) وحدات التخزين الثانوية

Secondary Storage Units

يسمى التخزين الذي يتم في الذاكرة الداخلية للحاسوب بالتخزين الأولي أو التخزين الداخلي . كما يسمى التخزين الذي يتم في أماكن التخزين الأخرى بالتخزين الثانوي أو الخارجي (External Storage) ، كما يطلق في كثير من الأحيان على وسائل التخزين الثانوية وسائل التخزين المساعدة (Auxiliary Storage) .

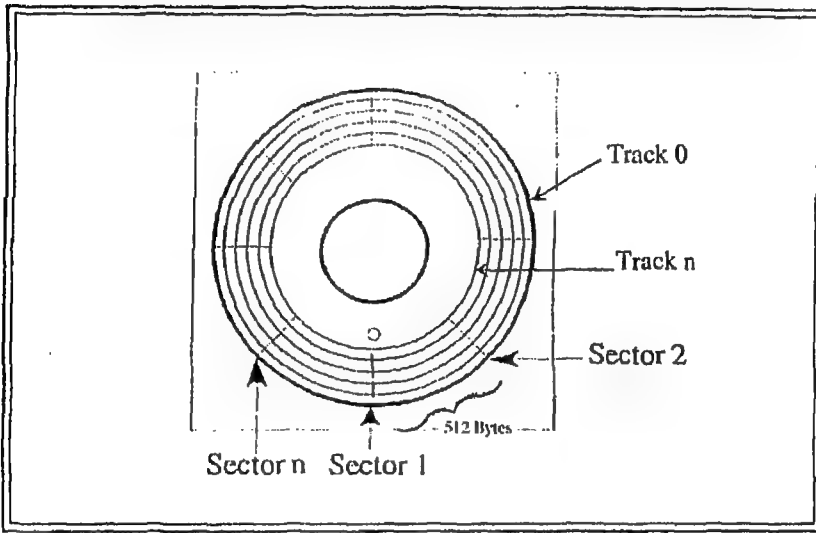
تعتبر الأقراص المغنطة (Magnetic Disks) أكثر وسائط التخزين شيوعاً واستخداماً ، وتتميز بإمكانية تخزين واسترجاع البيانات منها بطريقة مباشرة. من الناحية الشكلية يمكن تشبيه القرص بالاسطوانات التي تستخدم في تسجيل الاغاني والموسيقى، هناك اشكال مختلفة للأقراص المغنطة سوف نقوم بشرحها بشكل مفصل .

1- الأقراص الصلبة Hard Disks

تكون الأقراص الصلبة عادة على شكل حزمة من الأقراص Disk pack ، تتكون كل حزمة من عدد من الأقراص المعدنية الرقيقة الدائرية الشكل وكلا وجهيها مغطى بطبقة من مادة سريعة المغنطة . وقطر القرص هو 14 بوصة، يحتوي الوجه الواحد على عدة مئات من المسارات (Tracks) الدائرة لها نفس المركز تبدأ من حافة القرص ، ولكل منها رقم خاص ، يمثل كل حرف أو رقم بمجموعة من الأرقام الثنائية مرتبة على نفس المسار ، وعلى الرغم من اختلاف أطوال المسارات

إلا أن كل منها يسمح بتخزين نفس العدد من الحروف، مما يعني أن المسارات الخارجية تكون ذات درجة تركيز أقل من المسارات الداخلية والقريبة من مركز الدوران ، ينقسم كل مسار إلى عدد من القطاعات (Sectors) وترقم هذه القطاعات مثل المسارات . ويتم الوصول إلى مكان تخزين معين على القرص بطريقة مباشرة وذلك عن طريق تحديد رقم القطاع ورقم المسار .

الأقراص الشائعة الاستخدام لها 100 / 200 / 400 مسار، يتم ترقيم هذه المسارات من الخارج إلى الداخل انظر الشكل (7) .

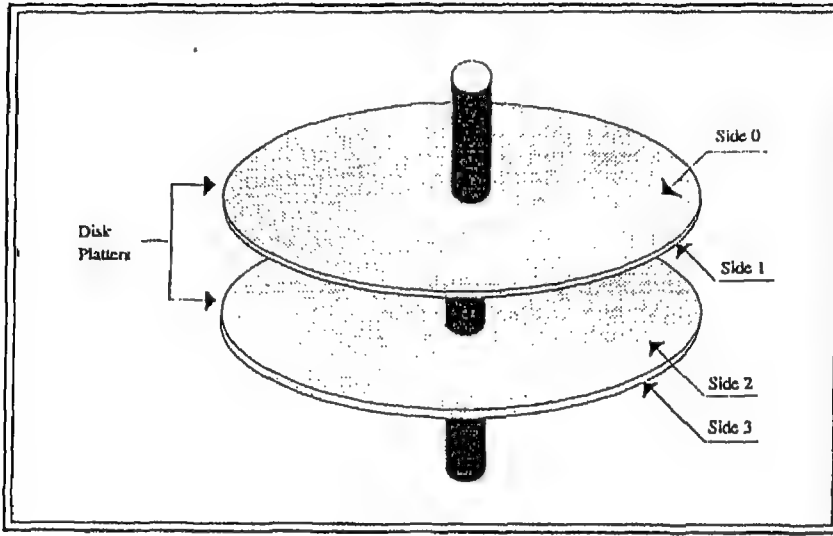


الشكل (7) يمثل قرص والمقاطع التي يتكون منها

يختلف عدد الأقراص في الحزمة من حاسوب إلى آخر ، كل قرص يسمى اسطوانة (Platter)

ومن أنواع حزم الأقراص الصلبة الشائعة الاستخدام حزمة أقراص ذات 5 اسطوانات وحزمة أقراص ذات 6 اسطوانات وهي الأكثر شيوعاً ، وحزمة أقراص ذات 21 اسطوانة ، وتتميز حزم الأقراص الممغنطة بسمات تخزين عالية.

وحزم الأقراص إما أن تكون مثبتة بصفة دائمة ، أو أن تكون قابلة للاستبدال بأخرى وليست جميع الأقراص مثبتة في حزم . فهناك أجهزة أقراص تستخدم قرصاً واحداً قابلاً للاستبدال ويسمى قرص الخرطوشة Disk Cartridge وبعضها يستخدم اثنين أو أكثر ، تبلغ سعة التخزين للخرطوشة من 40 إلى 150 ميغابت، وبالتالي فإن إمكانية تغيير الأقراص تعني إمكانية زيادة التخزين .



الشكل (8) الأقراص المركبة

عادة تكون الأقراص مركبة على محور دوار وبين كل اسطوانتين مسافة 0.5 بوصة وتدور الأقراص بسرعة عالية ، تعتبر السرعات 200 لفة / دقيقة أو 2400 لفة / دقيقة من السرعات الشائعة ويتم التسجيل على كلا الوجهين لكل قرص كتجمعات من النقاط المغنطة ، انظر الشكل (8) السابق .

يطلق على وحدات تشغيل الأقراص بالسواقات (Disk Drivers) والتي أحد مكوناتها رؤوس القراءة والكتابة ، هناك نوعان من وحدات رؤوس القراءة والكتابة على الأقراص المغنطة وهما الرأس المتحرك والرأس الثابت في حالة الرأس

المتحرك فإنه يتحرك أفقيا عبر سطح القرص بحيث يمكن الوصول إلى أي مسار محدد، هناك رأس لكل سطح ، وتتحرك جميع الرؤوس معاً ، اما في حالة الرؤوس الثابتة فهناك رأس للقراءة والكتابة مخصص لكل مسار ، وبالتالي فليس هناك حاجة لحركة الرأس كما أن الوصول إلى موقع تخزين محدد يتم بسرعة أكبر .

ليس هناك تلامس بين رؤوس القراءة والكتابة وأسطح الأقراص ، وإنما تركز هذه الرؤوس على وسائل من الهواء ، حيث ان حركة الهواء الناتجة من دوران الأقراص تجعل رؤوس القراءة والكتابة قريبة للغاية من سطح القرص . لذلك قد تسبب ذرات التراب في أتلاف بعض أجزاء من القرص ، كما تسبب في إفساد البيانات والمعلومات المخزنة عليها .

كيف يعمل القرص الصلب :

يمكن تلخيص عمل القرص الصلب وكما يلي :

1. يطلب التطبيق البرمجي من نظام التشغيل .
2. تستقبل البرامج الخدمية لذاكرة القرص طلب القراءة والكتابة للتأكد من أن البيانات المخزنة في الذاكرة الرئيسية مخصصة لذاكرة القرص الكاش . فإذا كانت البيانات المطلوبة موجودة في الذاكرة الكاش ، يتم نسخها إلى ذاكرة التطبيق الوسيطة Application Buffer ، وإذا لم تكن موجودة يحول طلب القراءة والكتابة إلى ضابط القرص الصلب لمعالجته .
3. يفحص ضابط القرص الصلب الذاكرة الوسيطة الموجودة على لوحة القرص لمعرفة فيما اذا كانت البيانات مخزنة هناك ، فإذا كانت كذلك تعاد البيانات إلى ذاكرة التطبيق الوسيطة .
4. اذا كانت البيانات غير مخزنة في ذاكرة القرص الصلب الوسيطة يقوم الضابط Controller بتشغيل المشغل الميكانيكي Actuator لرأس القرص الصلب.

5. يقوم مشغل رأس القرص بتحريك رؤوس القراءة والكتابة عبر الاطباق القرصية إلى المسار المعني .
6. يدور محور الدوران Spindle الموصل بسواعة محرك الاطباق القرصية لمساعد في جلب البيانات المطلوبة تحت رؤوس القراءة والكتابة .
7. تقوم رؤوس القراءة والكتابة بكتابة البيانات على الاطباق عن طريق تغيير القطبية .
8. يحتفظ ملف نظام التشغيل بسجل لمجموعة Clusters المستخدمة من قبل كل من ملف من الملفات .

2- أقراص ونشستر Winchester Disks

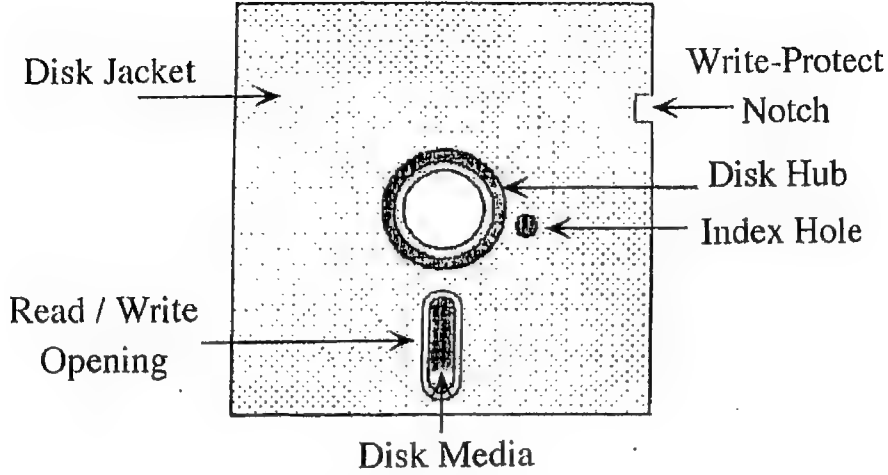
وهي عبارة عن أقراص محكمة العزل عن الهواء الخارجي ومصممة بحيث تستطيع رؤوس القراءة والكتابة الهبوط على سطح القرص . إذ يؤدي العزل عن الهواء الخارجي إلى منع التلوث بالغبار والدقائق الأخرى .

تعتبر أقراص ونشستر سريعة وقوية التحمل كما أنها منخفضة السعر مقارنة بالأقراص التقليدية الصلبة ، ونظراً لأنها محكمة الإغلاق فإنها لا تحتاج إلى صيانة دورية . الأحجام القياسية لهذه الأقراص هي 2,25 بوصة و 14 بوصة وهي مختلفة السعات التخزينية .

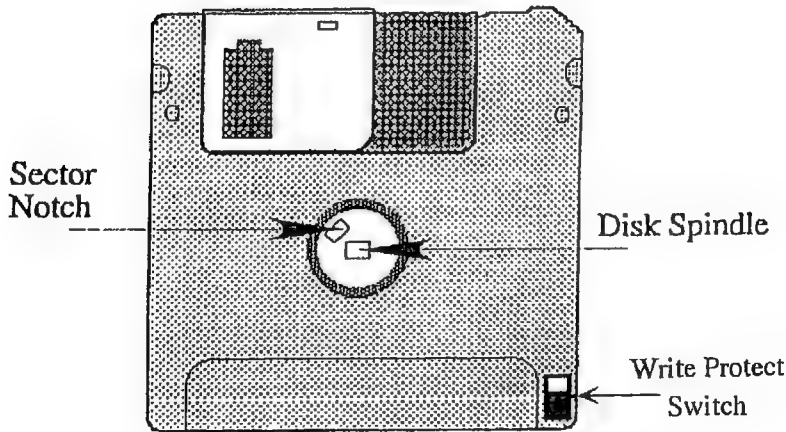
3- الأقراص المرنة Floppy Disks

يعتبر القرص المرن من وسائل التخزين الثانوية الشائعة الاستخدام وخاصة في الحواسيب الصغيرة ، والقرص مصنوع من نوع معين من البلاستيك وهو مغطى بمادة قابلة للمغنطة ومحفوظة داخل غلاف من البلاستيك وذلك لحماية السطح المغنط .

ينقسم كل قرص من الداخل إلى مجموعة من المسارات ، وكل مسار منها ينقسم إلى عدد من القطاعات تتواجد هذه الأقراص بأحجام مختلفة 5.25 بوصة و 3.5 بوصة ، كما في الشكلين (9) و (10) على التوالي .



الشكل (9) يمثل القرص المرن 5.25 بوصة



الشكل (10) يمثل القرص المرن 3.5 بوصة

يتم التسجيل على وجهي القرص أو على وجه واحد ، ترقيم المسارات على الأقراص المرنة القياسية من 0 الى 39 (40 مساراً) ، وترقم القطاعات من 1 الى 9 مما يعطي 360 قطاعاً إجمالياً (40 مساراً ضرب 9 قطاعات في المسار الواحد) على كل جانب ، ويتسع كل قطاع 512 بايت لذلك تبلغ سعة قرص مرن منخفض الكثافة على كلا الجانبين $(80 \times 18 \times 512 \times 2 = 1,474,560)$.
تمتلك الأقراص العالية الكثافة قياس 5.25 ، 80 مساراً في كل منها 15 قطاعاً ، مما يعطي 1200 قطاعاً وبالتالي تكون سعته 1.228.800 بايت (1,2MB) .
تحتوي الأقراص المرنة قياس 3.5 بوصة على 80 مساراً في كل جانب و 18 قطاعاً وبالتالي تكون سعتها 1,474,560 بايت (1.44 MB) بينما تحتوي الأقراص المرنة ذات 36 قطاعاً على 2,949,120 بايت (2,88 MB) .

4- أقراص الليزر الضوئية Laser Optical Disk

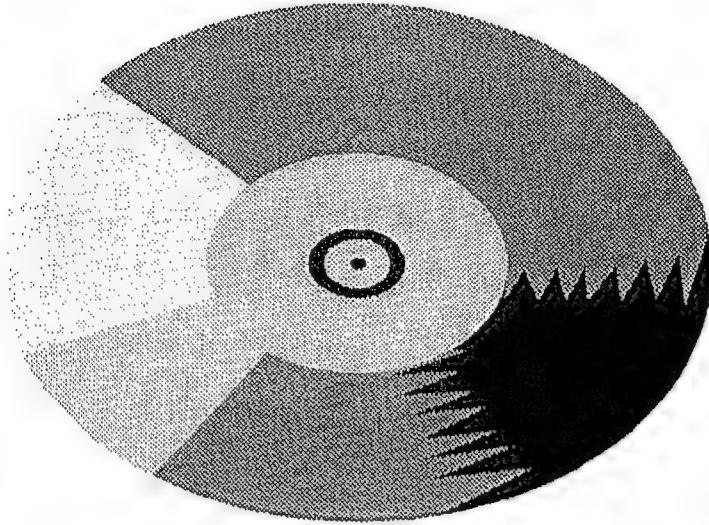
تعتبر أقراص الليزر الضوئية من أحدث أدوات التخزين الثانوية ، حيث تعتمد على تقنية تسجيل المعلومات من خلال أشعة الليزر الضوئية ، تستخدم هذه الأقراص لإدخال البرامج والمعطيات والبيانات الى الحاسوب ولكن التسجيل عليها يتم مرة واحدة وهذا ما ندعوه بالتسمية:

Read only memory – Copact Disk (CD)

تقوم هذه الأقراص بتخزين حجم هائل من البيانات يتراوح بين 440 MB وحتى 1.2 GB في الأقراص الحديثة .

القرص الليزري هو قرص دائري لا يزيد قطره على 4.51 أنش ، انظر الشكل (11) ، يتميز هذا القرص بأنه لا يمكن مسح البيانات المخزنة عليه ، كما يتميز بمقاومة أفضل العوامل البيئية وهو لا يتأثر بالغبار وأكثر تحملاً للحرارة والبرودة من الأقراص المرنة .

يتم التوجه حاليا لاستخدام أقراص ضوئية قابلة للمسح والتسجيل ، وقد وضعت بعض الشركات تصميم لشكل آخر من الأقراص دعتسه DVD ولكن هذه الأقراص ما تزال مرتفعة الثمن وقليلة الاستخدام .



الشكل (11) يمثل القرص الليزري

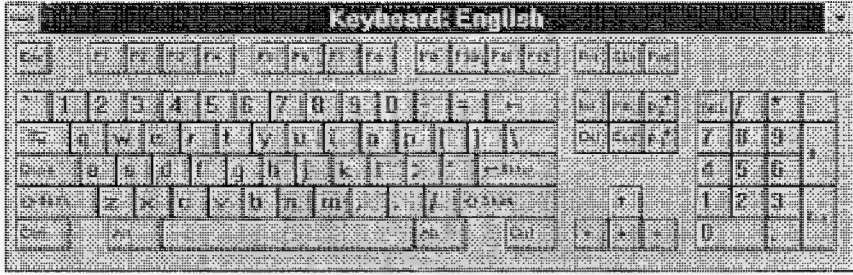
(2 - 5) وحدات الإدخال Input Units

تمثل وسائل الإدخال حلقة الربط بين مستخدمي الحاسوب ، والحاسوب نفسه ، حيث يستطيع مستخدم الحاسوب ادخال برامجه وبياناته إلى الحاسوب عن طريق وحدات ادخال متنوعة وتقوم وحدة الإدخال بتحويل الحروف والأرقام والعلامات الخاصة إلى ما يراها من الاعداد الثنائية ثم ارسالها إلى الذاكرة الرئيسية. سوف نوضح أهم وسائل الإدخال الشائعة .

1- لوحة المفاتيح Keyboard

وهي تشبه لوحة المفاتيح الخاصة بالالة الكاتبة إلى حد كبير وتعتبر أشهر وحدات الإدخال المستعملة الان . وهناك أنواع كثيرة مختلفة من لوحات المفاتيح ولكن لوحة المفاتيح القياسية والأكثر انتشارا هي تلك التي يتم توزيع الحروف والأرقام والعلامات الخاصة بها طبقا لما هو متبع في الآلة الكاتبة القياسية والتي يطلق

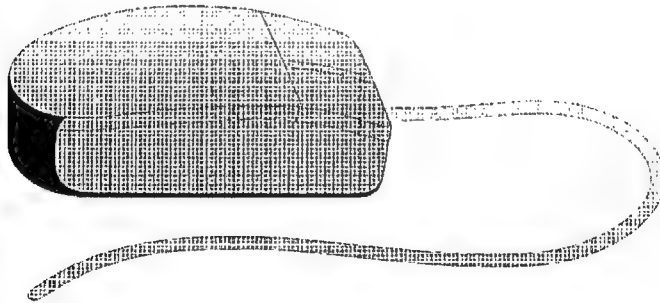
عليها لوحة المفاتيح كويرتي (Qwerty) والتي نشأ أسمها من مجموعته مفاتيح الحروف على الجانب الايسر من الصف الثالث من الاعلى . الشكل (12) تمثيل نموذجاً للوحة مفاتيح نموذجية لحاسب IBM .



الشكل (12) يمثل لوحة المفاتيح

2- الفأرة Mouse

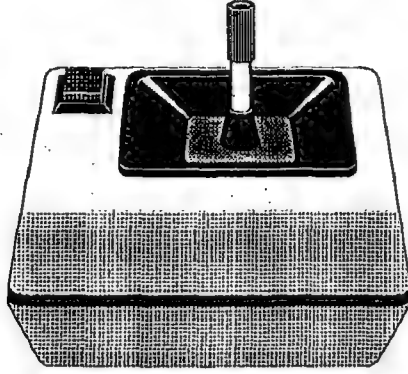
الفأرة عبارة عن جهاز تحكم إلكتروني في حجم قبضة اليد يرسل إشارات إلكترونية إلى الحاسوب وذلك لتحريك مؤشر الشاشة (cursor) . ويتم ذلك بتحريك الفأرة على أي سطح الكتب أو المنضدة الموضوع عليها الشاشة ، مما يؤدي إلى تحريك كرة دوارة موجودة أسفل الفأرة وبالتالي تحريك مؤشر الشاشة إلى الأمر الذي يكون مكتوباً أو مرسوماً على الشاشة وبالضغط على زر موجود على سطح الفأرة العلوي يتم اختيار هذا الأمر ويقوم الحاسب بتنفيذه . يمكن الاستفادة من الفأرة أيضاً في عملية الرسم . الشكل (13) يمثل أحد نماذج الفأرة .



الشكل (13)

3_ عصا التحكم اليدوي (Joystick)

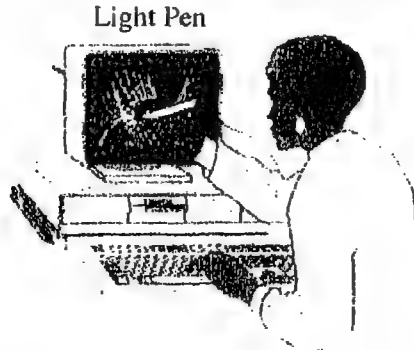
وتستخدم مع الألعاب بصفة خاصة لتوجيه أشياء معينة على الشاشة ، ويتم ذلك بواسطة تحريك ذراع صغير في الاتجاهات المختلفة . كما يوجد جهاز خاص يتم استخدامه في إطلاق القذائف ووسائل النيران المختلفة إذا كانت اللعبة تستدعي ذلك ، انظر الشكل(14) .



الشكل (14) يمثل نموذج لعصا التحكم اليدوي

4-القلم الضوئي (Light Pen)

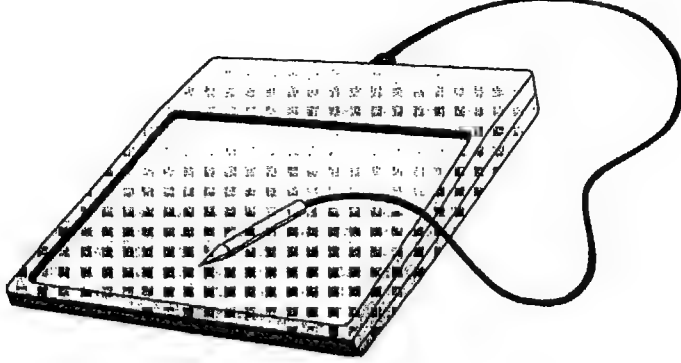
بملامسة هذا القلم لأي نقطة على الشاشة فان الحاسوب يقوم بقراءة موقع هذه النقطة ولذلك فهو يستخدم على نطاق كبير في التصميمات الهندسية وذلك لعمل تصميمات معقدة باستخدام الحاسوب وهو ما يطلق عليه تطبيقات تصميم بمساعدة الحاسوب (Computer Aided Design) واختصارا (CAD) .



الشكل (15) استخدام القلم الضوئي

5- لوحة الرسومات (Graphics Table)

وهي لوحة خاصة يمكن الرسم عليها وتنقل ما يرسم مباشرة الى شاشة الحاسوب وتستخدم عادة في التصميمات الهندسية ومكاتب الرسم . الشكل (16) يمثل نموذج للوحة الرسومات .



الشكل (16) يمثل لوحة الرسومات

6- أجهزة التعرف على الأصوات (Voice Recognition Devices)

استطاع بعض منتجي الحواسيب إنتاج أجهزة تمكن المستخدم من التحدث إلى الحاسوب . ويتم ذلك باستخدام ميكروفون ، الذي يقوم بتحويل الموجات الصوتية إلى موجات كهربائية يتم تحويلها إلى شفرات ثنائية ويتم مقارنتها بمعجم الألفاظ الصوتية ثم تسجيلية وتخزينه في ذاكرة الحاسوب . عندما يكون هناك توافق بين الكلمات الداخلة إلى الحاسوب مع تلك المخزنة في ذاكرته يقوم الحاسوب باجراء العمليات المطلوبة ، معظم أجهزة التعرف على الأصوات والتي تم انتاجها في السبعينات كانت تعتمد على متكلم معين ولا يستطيع الحاسوب التعرف على الكلمات الواردة من شخص آخر وذلك لاختلاف طبقات الصوت من شخص إلى آخر، في نهاية الثمانينات تم انتاج نظم عامة لا تعتمد طبقات صوت شخص معين، ولكن الكلمات التي يمكن التعرف عليها مازالت محدودة كما ان تكاليف هذه النظم ما زالت باهظة .

وقد قامت شركة IBM بتصميم نظام تعرف على الاصوات يمكن التعرف من خلاله على 6000 كلمة صوتية من تلك الكلمات الشائعة الاستعمال في المراسلات التجارية .

7- قارئ حروف الحبر المغنط

Magnetic Ink Character Reader (MICR)

يستخدم الحبر المغنط في كتابة الحروف والأعداد والرموز الخاصة على شيكات البنوك والمصارف ، ويتم ذلك باستخدام اشكال قياسية للحروف والأرقام والأشكال الخاصة والتي يتم طباعتها بأحبار تحتوي على رقائق قابلة للمغنطة من أكسيد الحديد ، يوجد هناك نظامين لاشكال الكتابة بنظام الحبر المغنط، النظام الأول هو نظام (E/3B) ويتكون من الارقام من صفر إلى تسعة بالإضافة إلى أربعة من الرموز الخاصة وهو يستخدم في الولايات المتحدة الأمريكية وإنكلترا ، ويستخدم هذا النظام في كتابة شيكات البنوك ، حيث يتم طباعة الشفرة الخاصة للبنك ورقم الحساب للعميل بالحبر المغناطيسي ، وعند تقديم الشيك إلى البنك يتم كتابة قيمة الشيك في الجانب الأيمن السفلي من الشيك بنفس الحبر المغنط وبواسطة آلة خاصة ، وتدخل بيانات الشيك إلى الحاسوب عن طريق وحدة المعالجة المركزية (CPU) بالحاسوب مباشرة أو تخزينها على شريط ممغنط بواسطة آلة خاصة وتشغيلها فيما بعد ، كما يمكن لوحدة القراءة أن تقوم بفرز تلك الشيكات على أساس اسم البنك أو رقم الحساب أو أي أساس آخر ووضعها في جيوب خاصة بآلة القراءة ، يبلغ متوسط سرعة قراءة الشيكات بهذا الاسلوب حوالي 1200-1500 شيك في الدقيقة الواحدة .

النظام الثاني للكتابة بنظام MICR هو نظام CMC7 ، وهو نظام نشأ أساساً في فرنسا ويستخدم في أوروبا ، وهو يحتوي على الأرقام من صفر إلى تسعة بالإضافة إلى الحروف الهجائية وخمسة رموز خاصة ، ويتم كتابة أي رمز منها باستخدام سبعة خطوط ممغنطة بينها ست مسافات متغيرة العرض ، المسافة الكبيرة تؤدي إلى

توليد الرقم الثنائي واحد ، بينما المسافة الصغيرة تؤدي إلى توليد الرقم الثنائي صفر.

تميز الحروف المطبوعة بالخير المغنط بأنه يمكن قراءتها دائما حتى بعد طمسها بالاختام في حالة شيكات البنوك ، وقارئ MICR يستطيع فقط التأكد من الشيك لكن عملية التأكد من المبلغ المسحوب وشخصية القوائم بالسحب والتأكد من صحة التوقيع مازالت أمور يتم القيام بها يدويا.

8- قارئ العلامات الضوئي

Optical Character Reader (OCR)

يوجد العديد من القارئات الضوئية منها ما هو ثابت على هيئة منضدة ، ومنها ما هو صغير الحجم يمكن الإمساك به باليد الواحدة وتدريبه على كود الاعمدة الموجودة على السلعة وبذلك يتم التعرف على السلعة ومواصفاتها وسعرها ان معظم أجهزة قراءة الحروف الضوئية تكون مصممة لكي تقرأ الحروف المطبوعة والمكتوبة على الآلة الكاتبة والمكتوبة بخط اليد .

9- الشاشة الحساسة للمس Touch Sensitive Screen

عن طريق لمس الشاشة يمكن توجيه الحاسوب لتنفيذ أوامر معينة .

(2-6) وحدات الإخراج Output Units

أجهزة الإخراج هي الأجهزة والوسائط التي تقوم باستقبال نتائج معالجة الحاسوب للبيانات الداخلة وتجهيزها بالصورة المطلوبة وعرضها على المستخدم ، سوف نلقي الضوء على أجهزة الإخراج الموجودة حالياً .

1- الشاشة Screen

ويطلق عليها أيضاً وحدات العرض المرئي Visual Display Units كما يطلق عليها اسم Visual Display Terminal ، وتعتبر من أشهر وسائل الإخراج المستخدمة ، كما تعتبر شاشة العرض من الأجزاء الرئيسية والهامة

للحاسوب إذ بواسطتها يمكن عرض وملاحظة النتائج التي يعالجها الحاسوب ويمكن أن يظهر على الشاشة نتائج رقمية أو نتائج جدولية أو رسوم وغيرها من الاشكال. يمكن أن يجهز الحاسوب بشاشة عادية أو شاشة رسم ، وتكون الشاشة العادية ذات ألوان أبيض وأسود ، يستخدم هذا النوع من الشاشات لعرض الأرقام والأحرف والرموز ، ويمكن أن تشكل بواسطة بعض الرموز الموجودة على لوحة المفاتيح جداول وأشكال حسب الإمكانيات ، يعتبر هذا النوع من الشاشات أقل ضرراً على الرؤية من النوع الثاني (شاشات الرسم) لذلك يفضل في كثير من الأحيان هذا النوع من الشاشات على شاشات الرسم ، أخيراً يطلق على هذا النوع من الشاشات اسم شاشات احادية اللون (Monochrome).

أما النوع الثاني فهو شاشات الرسم ويمكن أن تستخدم في كلا النظامين سواء كان رسم أو عرض نتائج كما في الشاشة العادية ، وهي في أغلب الأحيان ذات ألوان مختلفة يمكن التحكم بالألوان بواسطة أوامر خاصة بكل نظام من نظم البرمجة .

يكون حجم الشاشة أو قياسها في نظام عرض النتائج 25 سطر و 80 عمود وبذلك يمكن إظهار حوالي 2000 حرفاً على الشاشة ، أما في نظام الرسم فتكون ذات قياسات 600 عمود ، و 200 .

والجدول التالي يبين أنواع الشاشات الأكثر انتشاراً :

ألوان	نظام الرسم	ألوان	نظام عادي	نوع الشاشة
2	640x200	2	80x25	MDA (Monochrome Display Adapter)
16	640x200	16	80x25	CGA (Color/graphics Adopter)
2	720x340	2	80x25	Hercules
16	640x200	16	80x25	EGA(Enhanced Graphics Adapter)
256	640x200	256	80x25	MCGA(Multicolor Graphics Array)
256	640x480	256	80x25 80x50	VGA (Video Graphics Array)
256	640x480	256	80x25 80x50	SVGA(Super Video Graphics Array)

من أهم الصفات المميزة للشاشة سرعة عرض المعلومات في النظام العادي، كل الشاشات تقريبا تعمل بسرعة عالية لكن في نظام الرسم يختلف الأمر من شاشة الى أخرى حسب نوعها .

2- الطابعات Printers

هناك المئات من الطابعات المختلفة المتوفرة لعائلة الحواسيب الشخصية تؤدي جميعها نفس المهمة ولكن هناك أنواع مختلفة من الطابعات تعطيك درجات مختلفة من النوعية والأداء . والفئتان الرئيستان لطابعات الحواسيب الشخصية هي الطابعات النقطية (Dot Matrix) والطابعات الليزرية (Laser Printers) .

أ. الطابعات النقطية Dot Matrix

تعمل الطابعات النقطية بإطلاق سلسلة من الدبابيس المعدنية المرتبة في صف أفقي، تضغط هذه الدبابيس على شريط الحبر الذي ينشئ بدوره نقاط على الورق . وخلال تحريك رأس الطباعة (الأداة التي تحتوي على الدبابيس) ذهاباً وإياباً على الصفحة تقوم الدبابيس بإنشاء نمط معين هو عبارة عن صفوف من النقاط والتي تتألف منها الحروف .

حوالي عام 1985 زادت شركات التصنيع عدد الدبابيس في رأس الطباعة من 9 الى 24 . حيث وضعوا صفين من الدبابيس الصغيرة (12 دبوس في الصف الواحد) عوضاً عن صف واحد من الدبابيس الأكثر ثخانة ، وقد كانت النتيجة فعالة ومثيرة للإعجاب .

تستطيع الطباعة النقطية المؤلفة من 24 دبوس ان تعطي نوعية جيدة من الخط (Near Letter Quality) و اختصاراً (NLQ) وهي الحروف التي تقارب حروف الآلة الكاتبة من حيث الجودة .

تطبع الطباعة النقطية 80 حرف في السطر الواحد و 6 اسطر في البوصة ، كما تستطيع طباعة 132 حرف في السطر و 18 سطر في كل بوصة .

تستطيع الطابعة النقطية الجيدة ان تطبع 80 حرفا في الثانية الواحدة ، اما الطابعات الاسرع فبإمكانها طباعة حتى 160 حرفا في الثانية او حتى 240 حرفا في الثانية .

تجهز الطابعات النقطية بإحدى آليات تغذية الورق التالية :

- تغذية باسطوانة او التغذية الاحتكاكية .

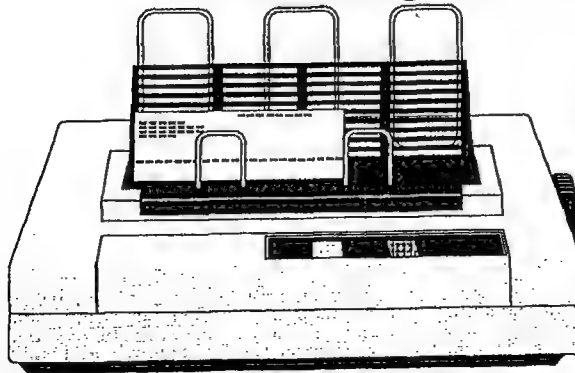
- تغذية بدبابيس .

- تغذية بجرار .

التغذية بالاسطوانة تقدم الورق مثلما يتم في الآلة الكاتبة ، وتقوم بوضع الورق في الطابعة وإدارة المقبض الى الأمام وضبط وضعية الورق تحت رأس الطابعة، وهذه الطريقة هي المثلى لطباعة صفحة واحدة او لطباعة المغلفات .

تستعمل آليات التغذية بدبابيس أو بجرار الصفحات المنطوية المتصلة في الطابعة ، وهذا الورق من النوع المزود بثقوب على الجانبين والتي تتطابق مع دبابيس الطابعة المستعملة لتحريك الورق في الطابعة .

أخيراً لا تزال الطابعات النقطية أكثر إنتشاراً بين طابعات الحواسيب وتختلف اسعارها حسب نوعية وسرعة الطابعة وعرض حامله راس الطابعة ، وكذلك حسب امكانية التحكم بألوان الطابعة وبنوع الخط ، كما تتعلق أسعارها بحجم ذاكرتها RAM التي تستعمل كمخزن وسيط Buffer لحفظ الأحرف الداخلة الشكل (17) يمثل نموذج لطابعة نقطية .



الشكل (17) يمثل طابعة نقطية

ب. الطابعات الليزرية Laser Printers

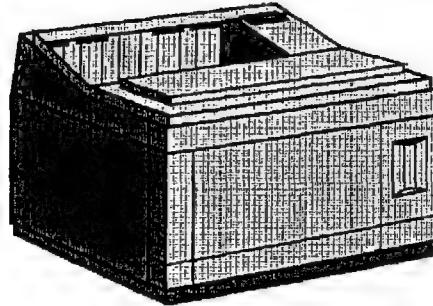
الطابعات الليزرية هي أكثر الطابعات قوة وأكثرها كلفة ، ولكن منذ عام 1989 تم تخفيض اسعار البعض منها .

تنشأ الصورة في الطابعة الليزرية بواسطة شعاع ليزر ، حيث تقوم دوائر إلكترونية معقدة بالتحكم بهذا الشعاع ، يرسم الشعاع صورة على الاسطوانة داخل الطابعة الليزرية بحيث تنشأ شحنة كهربائية حيثما يلامس الشعاع يلامس الشعاع الاسطوانة تدور الاسطوانة تحت خرطوشة حبر التصوير ملتقطة الحبر على اجزاء الاسطوانة المشحونة كهربائيا ، وعندما تلامس الاسطوانة الورق يتم نقل الصورة الى الورق ، وتستعمل الحرارة بعد ذلك لصهر وتثبيت حبر التصوير على الورق .

تمتلك الطابعة الليزرية ذاكرة RAM ، حيث تتعلق بها سرعة الطباعة فكلما زاد حجم الذاكرة كلما زادت سرعة الطباعة ، كما تقوم بحفظ البرامج التي يجب ان تطبع .

تتفوق الطابعة الليزرية على الطابعة النقطية أو أي نوع آخر من طابعات الحاسوب الشخصي وهي الطابعة المثالية التي تعطي نصا بجودة عالية .

لقد كانت الطابعات الليزرية الأولى تطبع أربع صفحات في الدقيقة ولكن الطرازات الحديثة تصل الى أكثر من 12 صفحة في الدقيقة الشكل (18) يمثل نموذج لاحدى الطابعات الليزرية .



الشكل (18) يمثل الطابعة الليزرية

3- الراسم Plotter

يقوم بإخراج النتائج في هيئة رسوم أو صور بيانية ، وهناك وحدات رسم تستخدم الأقلام (Pen Plotters) ومنها ما يستخدم اسطوانة (DRUM) وقاعدة مستوية (Flat Pen) ، وهناك وحدات رسم أخرى تستخدم راسمات الحبر النفث (Inkjet Plotters) وهي قادرة على عمل رسومات ذات حجم كبير وتستخدم الاسطوانة كحامل للورق .

يمكن للرسم ان يقوم برسم النتائج بالألوان وذلك باستخدام مجموعة من الأقلام الملونة او راسمات الحبر النفث الملونة . والراسم عادة بطيء ولكنه يتميز بدرجة عالية من الدقة وهي خاصية اهم من السرعة خاصة في التطبيقات الهندسية التي تتطلب دقة عالية .

4- الذراع الروبوتي Robotic Arm

تستخدم اذرع الروبوت حاليا بكثرة في معظم خطوط إنتاج المصانع الكبرى. يتم التحكم في الحركات الاساسية للذراع الروبوتي بواسطة الحاسوب، وتختلف طريقة وشكل الذراع طبقا لطبيعة وطريقة استخدامها .

بالإضافة إلى الأذرع الروبوتية فهناك العديد من الماكينات مثل ماكينات الخراطة والفريزة والتثقيب وماكينات الغزل والنسيج وخلافه يتم التحكم فيها بواسطة الحاسوب وهي تعتبر وحدات إخراج لنظام الحاسوب المستخدمة به .

5- الميكروفيلم Micro Film

يستخدم لتسجيل معلومات على شرائط افلام قياس 16 او 35 ملمتر بعد تصغير كل صفحة في الحجم بما يزيد عن 48 مرة.

6- التعامل مع الاصوات

ويتم ذلك باستخدام نظام الاستجابة للاصوات حيث يمكن استخدام الحاسوب لإجراء حوار او اتصال ، وتعتبر هذه الطريقة مناسبة إذا كان المطلوب مجرد معرفة إجابات تقليدية عن استفسارات حول معلومات معينة، ويتم ذلك عن طريقة إدخال الاستفسارات أو الرسائل وإرسالها عبر خطوط تليفونية الى حاسوب مركزي ، حيث يقوم نظام الإستجابة الصوتية بتركيب الاجابة من العبارات و الكلمات المسجلة و المخزنة مسبقاً ، ثم ترسل الاجابة الصوتية على الاستفسارات .

تمارين وأسئلة عامة

- 1- اذكر الوحدات الأساسية التي يتكون منها الحاسوب ؟
- 2- اذكر مكونات وحدة التشغيل المركزية ومكونات ووظيفة كل وحدة ؟
- 3- اشرح باختصار دورة المعالجة في وحدة التحكم ؟
- 4- اذكر أهم وحدات التخزين الداخلية ؟
- 5- ما هي مناطق الذاكرة الرئيسية ؟
- 6- ما هي أهم معايير التمييز بين الذاكرات ؟
- 7- تكلم عن الذاكر الفورية ؟
- 8- ما هي اشكال الذاكرة ROM ؟
- 9- عدد أنواع المسجلات ووظيفة كل منها ؟
- 10- تكلم عن المخازن الانتقالية ؟
- 11- اشرح آلية عمل القرص الصلب ؟
- 12- عدد أهم أنواع وحدات الإدخال ؟
- 13- عدد أهم أنواع وحدات الإخراج ؟
- 14- تكلم بإختصار عن القرص المرن 3.5 ؟

الفصل الثالث

الأنظمة العددية

Numbering Systems

(3 - 1) ماهية النظم العددية :

(3 - 1 - 1) مفاهيم النظم العددية :

استخدمت البشرية الأعداد منذ قدم الزمان ، فقد كان الراعي يحتاج الى نظام بسيط ليعد أغنامه فاستخدم أصابع يده العشرة أو ربما الحصى المتوفرة حوله ، واستخدم المصريون قبل الميلاد بـ 3400 سنة نظام العد العشري الذي نستخدمه نحن الآن ، ويعتقد أن سبب بناء نظام العد هذا على العدد عشرة ، يرجع الى أن بداية تعداد الأشياء كان باستخدام أصابع اليد العشرة .

إن النظام الذي اعتدنا الحساب به ليس الوحيد المستخدم في الوقت الحاضر فتوجد قبائل السكان الاصليين في استراليا وقبائل في بعض أجزاء غينيا الجديدة تستخدم النظام الثنائي ، أي نظام العد الذي فيه رقمين اثنين فقط ، إما صفراً أو واحد كما توجد قبائل في أمريكا الوسطى والمكسيك تستخدم نظام عد خماسي . وهذا ناتج عن احتياجهم لنظام عددي بسيط يناسب حضارتهم البدائية . يوجد أيضاً بعض القبائل في أمريكا الجنوبية والمكسيك تستعمل نظام عددي ذو أساس يتكون من عشرين رمزاً نتيجة لاهتمامهم بالفلك والنجوم .

يوجد هناك خمسة عشر نظاماً للعد الجدول (1) ، يبين بعضها لا يزال مستخدماً لحد الآن بينما البعض الآخر لا يتعدى كونه نظاماً مدرجاً في كتب الباحثين والمؤلفين . سوف نتعرض في هذا الفصل للأنظمة المستخدمة في الحواسيب.

أعداد النظام و رموزه	النظام بالعربية	النظام بالإنجليزية	الأساس
01	الثنائي	Binary	2
012	الثلاثي	Ternary	3
0123	الرباعي	Quaternary	4
01234	الخماسي	Quinary	5
012345	السداسي	Senary	6
0123456	السباعي	Septenary	7
01234567	الثماني	Octenary	8
012345678	التساعي	Nonary	9
0123456789	العشري	Denary	10
0123456789A	الاحد عشري	Undenary	11
0123456789AB	الأثنين عشري	Duodenary	12
0123456789ABC	الثالث عشري	Trenary	13
0123456789ABCD	الرابع عشري	Quatuordenry	14
0123456789ABCDE	الخامس عشري	Quinadenary	15
0123456789ABCDEF	السادس عشري	Hexadenary	16

جدول (1) الأنظمة العددية الخمسة عشر

(3 - 1 - 2) النظام العشري Dicimal System

يحتوي النظام العشري كما رأينا في الجدول أعلاه على عشرة أرقام أو رموز تمثل الأعداد الصحيحة من الصفر حتى التسعة، وعليه فإن أساس النظام العشري هو $b=10$ ، من المعروف أن أي عدد من النظام العشري يتكون من بعض أو كل القيم الموقعية التالية : أحاد ، عشرات ، مئات ، ...

ولإيجاد قيمة أي عدد نضرب كل رقم بقيمته الموقعية ، ثم نجمع نواتج الرموز المختلفة .

فمثلاً : العدد 6105 يتألف من :

الرقم 5 أحاد

الرقم 0 عشرات

الرقم 1 مئات

الرقم 6 آلاف

إذاً يمكن كتابته بالشكل التالي :

$$6105 = 6*1000 + 1*100 + 0*10 + 5*1$$
$$= 6*10^3 + 1*10^2 + 0*10^1 + 5*10^0$$

يدعى الشكل الأخير بالشكل الموسع (Expansion Form) ، حيث ضربنا كل رقم بقيمته الموقعية ، والقيمة الموقعية هي الأساس 10 مرفوعاً إلى الأس المساوٍ إلى موقع العدد :

الآحاد موقعه صفر ، العشرات موقعه 1 ، المئات موقعه 2 ، ...

وبناءً على ما سبق فإنه في أي نظام عددي يمكن كتابة العدد بالشكل

الموسع كما يلي :

نضرب كل رقم بقيمته الموقعية ثم نجمع نواتج الرموز المختلفة ، والقيمة الموقعية للرقم هو أساس النظام مرفوعاً للأس المساوٍ لموقع الرقم في العدد .

فإذا كان العدد مؤلف من n رقم على يسار الفاصلة (القسم الصحيح) و m رقم على يمين الفاصلة (القسم الكسري) عندئذ :

0 : يعبر عن موقع الرقم الأول الذي يأتي على يسار الفاصلة .

1 : يعبر عن موقع الرقم الثاني الذي يأتي على يسار الفاصلة .

2 : يعبر عن موقع الرقم الثالث الذي يأتي على يسار الفاصلة .

.....:

.....:

$n-1$: يعبر عن موقع الرقم n الذي يأتي على يسار الفاصلة .

-1 : يعبر عن موقع الرقم الأول الذي يأتي على يمين الفاصلة .

-2 : يعبر عن موقع الرقم الثاني الذي يأتي على يمين الفاصلة .

.....:

.....:

$-m$: يعبر عن موقع الرقم m الذي يأتي على يمين الفاصلة .

والصيغة العامة للشكل الموسع للعدد x في نظام عددي أساسه b هي :

$$X = (\dots d_3 d_2 d_1 d_0 , d_{-1} d_{-2} d_{-3} \dots)_b = \\ = \dots + d_3 b^3 + d_2 b^2 + d_1 b^1 + d_0 b^0 + d_{-1} b^{-1} + d_{-2} b^{-2} + d_{-3} b^{-3} \\ + \dots$$

حيث d يعبر عن الرقم في الموقع ، b أساس النظام .

مثال (1) : العدد 837.526 يعبر عنه بالشكل الموسع التالي :

$$837.524 = 8 \cdot 10^2 + 3 \cdot 10^1 + 7 \cdot 10^0 + 5 \cdot 10^{-1} + 2 \cdot 10^{-2} + 6 \cdot 10^{-3} \\ = 800 + 30 + 7 + 5/10 + 2/100 + 6/1000$$

(3-1-3) النظام الثنائي والعمليات الحسابية عليه Binary System

(1-3-1-3) مقدمة

إن العديد من العناصر الالكترونية للحاسوب ما هي الا عناصر ثنائية الاستقرار Bistable في طبيعتها ، بمعنى أنها تكون في احدى الحالتين فقط (كحالة سريان التيار فيها ON أو انقطاع التيار فيها OFF) ، إن هاتين الحالتين الممكنتين يشار اليهما بـ (0) و(1) والتي هي الرموز المستخدمة في النظام الثنائي، عدا عن ذلك فإن أي وحدة مستقلة من المعلومات تمثل عادة في الحاسوب بواسطة سلسلة من هذه الرموز الثنائية Binary Digits التي تدعى اختصاراً بالانجليزية BITS وهي اختصاراً للكلمتين حد ثنائي .

ان مثل هذه السلسلة يمكن النظر اليها على انها أعداد ثنائية وكثير من الحواسيب تستخدم نظام الأعداد الثنائية .

إذاً النظام الثنائي هو نظام يعتمد على الأساس $b=2$ إن الأعداد الثنائية

التي لا تحتوي على قسم كسري تدعى بالأعداد الثنائية الصحيحة Binary

Integers وقوى الأساس في النظام الثنائي للقسم الصحيح هي :

$$2^0, 2^1, 2^2, 2^3, \dots$$

وأن قوى الأساس للقسم الكسري هي :

$$2^{-1}, 2^{-2}, 2^{-3}, \dots$$

(2-3-1-3) التحويل من النظام الثنائي الى النظام العشري

Binary-Decimal Interconversion

لتحويل العدد في النظام الثنائي الى النظام العشري نتبع الخطوات التالية :

1. نكتب العدد بالشكل الموسع .
2. نوجد قيمة كل حد في الشكل الموسع
3. نوجد حاصل جمع الحدود .

مثال (2) :

$$(110101)_2 = 1*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0$$

$$= 32 + 16 + 0 + 4 + 0 + 1$$

$$= (53)_{10}$$

$$(101.1101)_2 = 1*2^2 + 0*2^1 + 1*2^0 + 1*2^{-1} + 1*2^{-2} + 0*2^{-3} + 1*2^{-4}$$

$$= 4 + 0 + 1 + 0.5 + 0.25 + 0 + 0.0625$$

$$= (5.8125)_{10}$$

والجدول (2) يحتوي على التمثيل الثنائي لبعض القيم الصحيحة و الجدول (3) يحتوي على قيمة الأساس المرفوع الى أس .

الأساس 2 مرفوع الى أس	القيمة العشرية	العدد العشري	العدد الثنائي
Power of Two	Decimal Number	Decimal Number	Binary Number
2^{10}	1024	0	0
2^9	512	1	1
2^8	256	2	10
2^7	128	3	11
2^6	64	4	100
2^5	32	5	101
2^4	16	6	110
2^3	8	7	111
2^2	4	8	1000
2^1	2	9	1001
2^0	1	10	1010
2^{-1}	$\frac{1}{2} = 0.5$	11	1011
2^{-2}	$\frac{1}{4} = 0.25$	12	1100
2^{-3}	$\frac{1}{8} = 0.125$	13	1101
		14	1110
		15	1111
		16	10000
		32	100000
		64	1000000
		128	10000000

الجدول (3)

الجدول (2)

(3 - 3 - 1 - 3) التحويل من العشري الى الثنائي

Decimal-Binary Interconversion

يمكننا إيجاد التمثيل الثنائي لرقم عشري N بتحويل جزئه الصحيح N_I

وجزئه الكسري N_F كل جزء بصورة مستقلة . سنوضح العملية بالمثال التالي :

مثال (3) : حول العدد العشري $N = 109.78125$ الى مكافئة الثنائي .

أ- نحول القسم الصحيح $N_I = 109$ الى مكافئة الثنائي ، من أجل ذلك نقسم

N_I على الأساس 2 ونكتب القسم الصحيح من ناتج القسمة في عامود نتائج

القسمة ، والباقي نضعه في عامود البواقي ثم نقسم حاصل القسمة على 2 ...

وهكذا مع تسجيل الباقي بعد كل عملية تقسيم على اليمين .

البواقي Remainders	ناتج القسمة Quotients	التقسيمات Divisions
1	54	$109 \div 2$
0	27	$54 \div 2$
1	13	$27 \div 2$
1	6	$13 \div 2$
0	3	$6 \div 2$
1	1	$3 \div 2$
1	0	$1 \div 2$

نلاحظ ان بواقي القسمة لا يمكن إلا أن تكون مساوية للواحد أو الصفر ما دامت

عملية التقسيم تتم على 2 .

إن سلسلة البواقي مقروءة من الأسفل الى الأعلى (كما يشير السهم) تعطي

المكافئ الثنائي المطلوب ، و بالتالي :

$$N_I = (109)_{10} = (1101101)_2$$

ب- أما لتحويل القسم $N_F = 0.78125$ الى مكافئة الثنائي فيتم بضرب N_F بالأساس 2 ونأخذ من ناتج الضرب القسم الصحيح الذي سيكون إما صفراً أو واحد ونضعه في عامود الأقسام الصحيحة ، ثم نأخذ الجزء الكسري من ناتج الضرب ونضربه بـ 2 ، وهكذا نعيد ما سبق .

الاقسام الصحيحة	عمليات الضرب
Integral Parts	Multiplications
1	$0.78125 \times 2 = 1.56250$
1	$0.5625 \times 2 = 1.1250$
0	$0.125 \times 2 = 0.250$
0	$0.250 \times 2 = 0.50$
1	$0.50 \times 2 = 1.00$

إن القسم الكسري الصفري في ناتج الضرب بـ 2 يشير الى نهاية الحسابات . نلاحظ أن القسم الصحيح لأي جداء لا يمكن أن يكون إلا صفراً أو واحداً ما دمنا نضاعف رقماً أقل من الواحد إن سلسلة حدود القسم الكسري من الأعلى إلى الأسفل كما هو مشار إليها بالسهم تعطي المكافئ الثنائي المطلوب . أي أن :

$$N_F = (0.78125)_{10} = (0.11001)_2$$

وعليه فإن المكافئ الثنائي الكلي للرقم العشري N هو مجموع المكافئين أي :

$$N = N_I + N_F$$

$$(109.78125)_{10} = (1101101.11001)_2$$

(3-1 -3-4) العمليات على النظام الثنائي

أولاً : جمع الأعداد الثنائية Binary Addition

تجمع الأعداد الثنائية وفق القواعد المبينة في الجدول التالي :

$0+0 = 0$	
$0+1 = 1$	
$1+0 = 1$	
$1+1 = 0$	؛ Carry 1 مع نقل باليد واحد
$1+1+1 = 1$	؛ Carry 1 مع نقل باليد واحد

مثال (4) : احسب المجموع الثنائي للعددين الثنائيين التاليين :

$$110011101 + 10110111$$

الحل : لنكتب العددين على الشكل التالي :

$$\begin{array}{r} 110011101 \\ + \\ 10110111 \\ \hline 1001010100 \end{array}$$

مثال (5) : احسب المجموع الثنائي للأعداد الثنائية الاربعة التالية :

$$1001 + 1101 + 110 + 1011$$

الحل : نجمع العددين الأول و الثاني ثم نجمع الى المجموع الناتج العدد الثالث ثم

نجمع الى المجموع الناتج العدد الرابع فتحصل على القيمة الاجمالية أي :

$$\begin{array}{r} 1001 \\ + 1101 \\ \hline 10110 \\ + 110 \\ \hline 11100 \\ + 1011 \\ \hline 100111 \end{array}$$

مثال (6) : احسب المجموع الثنائي التالي :

$$11011.01 + 101.1101$$

الحل : نكتب الأعداد الثنائية بحيث تقع الفواصل الثنائية التي تفصل القسم الكسري

عن القسم الصحيح تحت بعضها ثم نجمع :

$$\begin{array}{r} 11011.01 \\ + \quad 101.1101 \\ \hline 100001.0001 \end{array}$$

ثانياً : ضرب الأعداد الثنائية Binary multiplication

يتم ضرب الأعداد الثنائية وفق القواعد المبينة في الجدول التالي :

$0 * 0 = 0$
$0 * 1 = 0$
$1 * 0 = 0$
$1 * 1 = 1$

مثال (7) : أوجد جداء العددين التاليين :

$$\begin{array}{r} 1101011 \\ * \quad 10110 \\ \hline 0000000 \\ 1101011 \\ 1101011 \\ 0000000 \\ +1101011 \\ \hline 100100110010 \end{array}$$

مثال (8) : احسب الجداء الثنائي التالي $11.01 * 101.1$:

الحل :

$$\begin{array}{r}
 11.01 \\
 * 101.1 \\
 \hline
 1101 \\
 1101 \\
 0000 \\
 \underline{1101} \\
 10001.111
 \end{array}$$

نلاحظ وجود ثلاثة أرقام بعد الفاصلة وبالتالي هي نفس الطريقة المتبعة في النظام العشري .

ثالثاً : الطرح الثنائي Binary Subtraction

يتم طرح الأعداد الثنائية وفق القواعد المبينة في الجدول التالي :

$0 - 0 = 0$
$1 - 0 = 1$
$1 - 1 = 0$
$0 - 1 = 1$; (Borrow 1) مع الاستعارة واحد من العمود التالي

القاعدة الأخيرة تأتي من

$$10 - 1 = 1 \text{ وكذلك } 1 + 1 = 10$$

وبالتالي الفرق $0 - 1$ يتطلب الاستعارة التي تعطي $10 - 1 = 1$

مثال (9) : أوجد حاصل طرح الأعداد التالية :

$$11101 - 1011 =$$

الحل : نكتب الأعداد بحيث تكون الأعداد مرتبة في كلا العددين تحت بعضها

البعض ابتداءً من جهة اليمين ، أي على الشكل التالي :

$$\begin{array}{r} 11101 \\ - 1011 \\ \hline 10010 \end{array}$$

مثال (10) : أوجد حاصل طرح الأعداد التالية :

$$1100101001 - 110110110 =$$

الحل :

$$\begin{array}{r} 1100101001 \\ - 110110110 \\ \hline 101110011 \end{array}$$

نلاحظ في العمود الثاني وفي العدد الأول وجود صفر وفي العمود الثالث أيضاً وجود صفر لذلك سوف نستعير الواحد من العمود الرابع وبالتالي تصبح قيمة العمود الثالث الذي اخترناه أثناء الاستعارة مساوياً للواحد وتصبح قيمة العمود الرابع صفر والعمود الثاني 10 .

مثال (11) : أوجد حاصل طرح الأعداد الثنائية التالية :

$$1101.101 - 11.10111 =$$

الحل : يجب أن نضع الأعداد الثنائية في كلا العددين بحيث تقع الفواصل تحت بعضها البعض ، أي :

$$\begin{array}{r} 1101.10100 \\ - 11.10111 \\ \hline 1001.11101 \end{array}$$

لا حظ أننا أضفنا في العدد الأول صفيرين الى الجزء الكسري على يمين الفاصلة وذلك من أجل أن يصبح عدد الأعداد على يمين الفاصلة في العدد الأول مساوياً لعدد الأعداد على يمين الفاصلة في العدد الثاني .

رابعاً : التقسيم الثنائي Binary Division

القواعد المتبعة في تقسيم الأعداد الثنائية هي نفسها المتبعة في تقسيم الأعداد العشرية سواء أكانت صحيحة أم كسرية .

مثال (12) : أوجد ناتج ما يلي :

$$101001 \div 11 =$$

الحل :

$$\begin{array}{r}
 \begin{array}{c} \text{ناتج القسمة} \leftarrow 11011 \\ \leftarrow \text{المقسوم} \end{array} \\
 \begin{array}{r}
 \begin{array}{c} \text{المقسوم عليه} \rightarrow 11 \end{array} \overline{) 1010001} \\
 \underline{- 11} \\
 100 \\
 \underline{- 11} \\
 0100 \\
 \underline{- 11} \\
 011 \\
 \underline{- 11} \\
 00 \leftarrow \text{باقي القسمة}
 \end{array}
 \end{array}$$

وبالتالي حاصل القسمة هو العدد : 11011 : أي أن :

$$1110111 \div 11 = 11011$$

مثال (13) : أوجد ناتج ما يلي :

$$1110111 \div 1001 =$$

الحل :

$$\begin{array}{r}
 1101 \\
 1001 \overline{) 1110111} \\
 \underline{- 1001} \\
 1011 \\
 \underline{- 1001} \\
 10
 \end{array}$$

إذاً حاصل القسمة هو 101 والباقي 10 أي أن :
 $1110111 \div 1001 = 1101$

و الباقي 10 .

مثال (14) : أوجد ناتج ما يلي :

$$111.00001 \div 1.01 =$$

الحل : ننقل الفاصلة الثنائية بين القسم الصحيح والكسري في كل من القاسم والمقسوم عليه بمقدار منزلتين لتحويل القاسم 1.01 الى عدد صحيح ثم نجري عملية القسمة :

$$\begin{array}{r}
 101.101 \\
 101 \overline{) 11100.0001} \\
 \underline{- 101} \\
 1000 \\
 \underline{- 101} \\
 101 \\
 \underline{- 101} \\
 000
 \end{array}$$

وبالتالي نجد ان حاصل القسمة 101.101 أي :

$$111.00001 \div 1.01 = 101.101$$

(3 - 1 - 4) نظام العد الثماني Otcal System

(3 - 1 - 4 - 1) مقدمة

إن نظام العد الثماني له أساس $b=8$ وأعداده الثمانية هي :

0 , 1 , 2 , 3 , 4 , 5 , 6 , 7

وهذا يختلف عن نظام العد العشري باختفاء الرموز 8 و 9 وبذلك يأتي بعد العدد 7 العدد 10 وبعده يأتي العدد 11 وهكذا حتى العدد 17 حيث يليه العدد 20 ومن ثم 21 حتى 27 وهكذا ... وبالتالي نجد ان التعداد باستعمال رموز النظام العد الثماني هو كالآتي :

0,1,2,3,4,5,6,7

10,11,12,13,14,15,16,17

20,21, 22,23,24,25,26,27

30,31,

(3-1-4-2) التحويل من النظام الثماني الى النظام العشري

Octal-Decimal Interconversion

لقد لاحظنا غياب العددين 8 و 9 من أعداد النظام الثماني وأن العدد 30 في القائمة أعلاه يعني العدد 24 في النظام العشري ، حيث أن كل موقع فيه يساوي ثمانية مرات الموقع الذي يقع على يمينه، سنأخذ عدد الثماني 30 ونطبق عليه مبدأ التحويل إلى النظام العشري حيث نضرب العدد 0 بـ 8^0 والعدد 3 بالعدد 8^1 ثم نجمع الحدين نجد :

$$(30)_8 = 3 \cdot 8^1 + 0 \cdot 8^0 = 3 \cdot 8 + 0 \cdot 8 = (24)_{10}$$

بعبارة أخرى : لتحويل العدد في النظام الثماني الى النظام العشري نتبع الخطوات التالية :

1. نكتب العدد بالشكل الموسع .
2. نوجد قيمة كل حد في الشكل الموسع بالنظام العشري .

3. نوجد حاصل جمع الحدود .

مثال (15) : حول العدد الثماني 3516 الى ما يكافئه في النظام العشري

$$(3516)_8 = 3 \cdot 8^3 + 5 \cdot 8^2 + 1 \cdot 8^1 + 6 \cdot 8^0 = 3 \cdot 512 + 5 \cdot 64 + 1 \cdot 8 + 6 \cdot 1$$
$$= 1536 + 320 + 8 + 6 = (1870)_{10}$$

أي ان العدد $(3516)_8$ يكافئ $(1870)_{10}$ ، حيث ان العدد 8 هو أساس

النظام الثماني والعدد 10 هو أساس النظام العشري .

(3-4-1-3) التحويل من النظام الثماني الى النظام الثنائي وبالعكس

Octal-Binary Interconversion

هناك علاقة قوية بين أساس النظام الثنائي وأساس النظام الثماني وهذه

العلاقة هي :

$$8 = 2^3$$

هذا يعني أنه يمكن كتابة كل رقم في النظام الثماني بما يكافئه في النظام

الثنائي، وذلك بتبديل كل رقم ثماني بثلاثة ثنائيات، و العكس صحيح، كما في

الجدول التالي :

الرقم الثماني OCTAL DIGITS	المكافئات الثنائية BINARY EQUIVALENTS
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

ملاحظة : عند التحويل من النظام الثنائي الى النظام الثماني، نقوم بأخذ ثلاثة أعداد

من اليمين الى اليسار ونستبدلها بما يكافئها إذا كان العدد صحيحاً ، أما

إذا كان يحتوي على قسم كسري عندئذ نأخذ كل ثلاثة أعداد ابتداءً من الفاصلة الثنائية التي تفصل القسم الكسري عن القسم الصحيح ونستبدلها بما يكافئها في النظام الثماني .

مثال (16) : حول العدد الثماني 3703 الى مكافئه في النظام الثنائي .

الحل :	3	7	0	3
	011	111	000	011

و بالتالي نجد $(11111000011)_2 = (3702)_8$

مثال (17) : حول العدد $(10101011111)_2$ الى مكافئه الثماني :

الحل : نقطع العدد الثنائي الى قطع ثلاثية ابتداءً من اليمين ثم استبدال كل قطعة بما يكافئها في النظام الثماني :

010	101	011	111
2	5	3	7

وبالتالي فإن العدد $(2537)_8$ هو المكافئ الثماني للعدد $(10101011111)_2$
 لاحظ أننا أضفنا صفرًا للقطعة الأخيرة حتى أصبحت ثلاثية .

مثال (18) : حول العدد $(1101.0101)_2$ إلى مكافئه الثماني .

الحل : نقطع العدد الثنائي الى قطع ثلاثية ابتداءً من يسار ثم يمين الفاصلة الثنائية ثم نستبدل كل قطعة بمكافئها الثماني .

011	011	010	100
3	3	2	4

و بالتالي فإن :

$$(11011.0101)_2 = (33.23)_8$$

(3-1-4-4) العمليات في النظام الثماني :

أولاً: الجمع في النظام الثماني Octal Addition

يتم جمع الأعداد الثمانية وفق الجدول التالي :

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

و لصعوبة حفظ الجدول السابق فإنه يمكن استخدام الأسلوب التالي لإيجاد مجموع

عددين في النظام الثماني :

1. نرتب العددين تحت بعضهما البعض ابتداءً من اليمين .
2. نبدأ من العمود الأول الواقع في الجهة اليمين .
3. نجمع العددين في العمود كما نجمع في النظام العشري .
4. إذا كان الناتج أكبر من (7) نطرح منه العدد (8) ونضع العدد (1) فوق العمود التالي الذي يقع على يساره .
5. نطبق الخطوتين (3 و 4) على العمود التالي مع الانتباه الى انه يجب جمع العدد (1) مع أعداد العمود إذا كان هذا الرقم موجوداً نتيجة الخطوة (4) .

مثال (19) : اجمع كل من التراكيب التالية :

- a) $(5)_8 + (4)_8$, d) $(7)_8 + (4)_8$, g) $(1)_8 + (5)_8 + (6)_8$
b) $(6)_8 + (7)_8$, e) $(1)_8 + (4)_8 + (2)_8$
c) $(3)_8 + (2)_8$, f) $(1)_8 + (6)_8 + (3)_8$

الحل :

a	b	c	d	e	f	g
				1	1	1
5	6	3	7	4	6	5
+ 4	+ 7	+ 2	+ 4	+ 2	+ 3	+ 6
9	13	5	11	7	10	12
- 8	- 8	- 0	- 8	- 0	- 8	- 8
11	15	5	13	7	12	14

نلاحظ أن المجموع العشري الذي يزيد عن السبعة نطرح منه العدد 8 ونضع على يسار الناتج العدد واحد ، و ان المجموع العشري الذي لا يزيد عن سبعة نطرح منه العدد صفر ويبقى الناتج على حالة .

مثال (20) : أوجد المجموع الثماني التالي :

$$(7346)_8 + (5263)_8 =$$

الحل : من اجل ايجاد المجموع نرتب العددين بالطريقة العادية ونطبق على كل عمود وبصورة مستقلة قاعدة جمع الأعداد في النظام العشري ، لاحظ أن المحمولات التي تظهر من جراء طرح العدد 8 في الخطوة الخاصة بالتعديل ينقل من اسفل العمود إلى أعلى العمود التالي الواقع إلى اليسار .

	1		1	1	
		7	3	4	6
+		5	2	6	3
المجموع العشري		12	6	11	9
التعديل		-8	-0	-8	-8
المجموع الثماني	1	4	6	3	1

و بالتالي العدد الناتج هو $(14631)_8$ أي :

$$(7346)_8 + (5263)_8 = (14631)_8$$

ثانياً : الطرح في النظام الثماني Octal Subtraction

لإيجاد الفرق الثماني $y = B - A$ نوجد أولاً المتمم السباعي للرقم A (العدد المطروح) حيث نطرح كل رقم فيه من العدد 7، ثم نجري عملية الجمع العادي التي مرت معنا سابقاً وأخيراً نقوم بحذف الرقم الأخير من الناتج وهو العدد 1 .

مثال(21) : أوجد الفرق الثماني للأعداد $A = 3141$ ، $B = 7526$

$$Y = B - A \text{ حيث}$$

الحل : أ. نوجد المتمم السباعي لـ A و ذلك بطرح كل رقم من أرقام A من 7.

متمم A هو : 4636

ت- نجمع متمم A إلى B

1				1		
	7	5	2	6	B	
+	4	6	3	6	A	متمم
<hr/>						
	12	11	6	12	الجمع العشري	
	-8	-8	-0	-8		
①	4	3	6	4		

ج — نحذف العدد (1) الموجود على يسار الناتج نجد الفرق المطلوب

$$y = (4364)_8$$

(3 - 1 - 5) النظام السادس عشر Hexadecimal System

(3-1-5-1) مقدمة

النظام السادس عشر يستعمل ستة عشر رقماً وهي الأرقام من الصفر حتى التسعة مضافاً إليها ست حروف هجائية وفق ما هو موضح في الجدول التالي :

المكافئات في النظام الثنائي Binary Equivalents	القيم المقابلة في النظام العشري Decimal Values	أرقام النظام السادس عشر Hexadecimal Digits
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

(3 - 1 - 5 - 2) التحويل من النظام السادس عشر الى النظام العشري

وبالعكس Interconversion Hexadecimal – Decimal

لتحويل العدد في النظام السادس عشر إلى مكافئه في النظام العشري نتبع مايلي :

سوف نوضح ذلك من خلال الامثلة التالية :

- 1- نكتب العدد السادس عشر بالشكل الموسع .
- 2- نوجد قيمة لكل حد في الشكل الموسع بالنظام العشري .
- 3- نوجد حاصل جمع الحدود .

مثال (22) : حول العدد $(73D5)_{16}$ الى مكافئه العشري :

الحل : نعلم ان الحرف $D = 13$ وبالتالي :

$$\begin{aligned}(73D5)_{16} &= 7*16^3 + 3*16^2 + 13*16^1 + 5*16^0 \\ &= 7*4096 + 3*256 + 13*16 + 5*1 \\ &= 28672 + 768 + 208 + 5 = (29653)_{10}\end{aligned}$$

مثال (23) : حول العدد السادس عشر $(39.B8)_{16}$ الى مكافئه العشري .

الحل :

$$\begin{aligned}(39.B8)_{16} &= 3*16^1 + 9*16^0 + 11*16^{-1} + 8*16^{-2} \\ &= 3*16 + 9*1 + 11* 0.0625 + 8*0.0039062 \\ &= 48 + 9 + 0.6875 + 0.03125 = (57.71875)_{10}\end{aligned}$$

لتحويل العدد من النظام العشري الى مكافئه في النظام السادس عشر نتبع

نفس الخطوات التي استخدمناها لتحويل العدد من النظام العشري الى النظام الثنائي مع الأخذ بعين الاعتبار أن الأساس هنا هو 16 ، و الأمثلة التالية توضح هذه العملية.

مثال (24) : حول العدد العشري $P = (9719)_{10}$ الى مكافئه السادس عشر .

الحل : من أجل التحويل من النظام العشري الى النظام السادس عشر نقسم P وكل حاصل قسمة لاحق على الأساس 16 مع ملاحظة البواقي كما في الجدول التالي :

البواقي Remainders	ناتج القسمة Quatients	التقسيمات Divisions
7	607	$9719 \div 16$
15	37	$607 \div 16$
5	2	$37 \div 16$
2	0	$2 \div 16$

ان سلسلة البواقي التي نبدل فيها الباقي العشري 15 بالعدد السادس عشر F ، مقروءة بالترتيب المعكوس ، هي التي تعطي السادس عشر للعدد $P=9719$ أي أن :

$$(9719)_{10} = (25F7)_{16}$$

مثال (25) : حول الكسر العشري $Q = 0.78125$ إلى مكافئه السادس عشر .

الحل : من أجل تحويل الكسر العشري الى مكافئه السادس عشر نقوم بضرب الكسر بالأساس 16 حتى تتم عملية تحويل الكسر الى عدد صحيح .

الأقسام الصحيحة Integer Parts	عمليات الضرب Multiplications
12	$0.78125 * 16 = 12.50000$
8	$0.50000 * 16 = 8.00000$

في هذه الحالة نحصل على قسم كسري صفري . إن سلسلة الأقسام الصحيحة والتي نبدل فيها العدد العشري 12 بمكافئه السادس عشر C هي التي تعطي الشكل السادس عشر المطلوب للعدد Q أي :

$$(0.78125)_{10} = (0.C8)_{16}$$

(3-5-1-3) التحويل من النظام السادس عشر الى النظام الثنائي وبالعكس

Hexadecimal-Binary Interconversion

هناك علاقة بين أساس النظامين الثنائي والسادس عشر كما يلي :
أي أنه يمكن كتابة كل عدد في النظام السادس عشر بما يكافئه في النظام الثنائي، وذلك بتبديل كل عدد ست عشري بأربعة ثنائيات ، والعكس صحيح .

مثال : ما هو المكافئ السادس عشر للعدد الثنائي التالي :

1010001011010010

الحل : نكتب العدد الثنائي على الشكل التالي :

1010 0010 1101 0010

A 2 D 2 و هذا يكافئ

و بالتالي العدد السادس عشر المكافئ هو $(A2D2)_{16}$

مثال (27) : حول العددين $(DC10)_{16}$ ، $(B747)_{16}$ الى مكافئهما في النظام الثنائي .

الحل : نعوض كل رقم من الأعداد السابقة بما يكافئه من الأعداد الثنائية

و بالتالي :

$$(DC10)_{16} = (1101\ 1100\ 0001\ 0000)_2$$

$$(B747)_{16} = (1011\ 0111\ 0100\ 0111)_2$$

(3-1-5-4) العمليات الحسابية في النظام السادس عشرة

أولاً : الجمع في النظام السادس عشر Hexadecimal Addition

يتم جمع الأعداد الستة عشرية وفق الجدول التالي :

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

ولصعوبة حفظ الجدول السابق لاستخدامه عند جمع الأعداد فإنه يمكن استخدام الطريقة التالية :

1. ترتيب العددين المراد جمعهما تحت بعضهما البعض ابتداءً من اليمين .
2. نبدأ من العمود الأول الواقع جهة اليمين .
3. نجمع الرقمين في العمود كما نجمع في النظام العشري .
4. اذا كان الناتج أكبر من 15 نطرح منه العدد (16) ونضع العدد (1) فوق العمود التالي الذي يقع على يساره .
5. نطبق الخطوتين (3و4) على العمود التالي مع الانتباه إلى أنه يجب جمع العدد (1) مع أرقام العمود إذا كان هذا العدد موجوداً نتيجة الخطوة (4) .

مثال (28) : أوجد ناتج جمع كل من التراكيب التالية في النظام السادس عشر :

- a) $(8)_{16} + (9)_{16}$, e) $(C)_{16} + (D)_{16}$, i) $(1)_{16} + (5)_{16} + (C)_{16}$
b) $(3)_{16} + (5)_{16}$, f) $(3)_{16} + (B)_{16}$, j) $(1)_{16} + (E)_{16} + (6)_{16}$

- c) $(6)_{16} + (7)_{16}$, g) $(F)_{16} + (D)_{16}$
d) $(A)_{16} + (9)_{16}$, h) $(1)_{16} + (4)_{16} + (6)_{16}$

a	b	c	d	E	f	G	h	i	j
							1	1	1
8	3	6	A	C	3	F	4	5	E
<u>+9</u>	<u>+5</u>	<u>+7</u>	<u>+9</u>	<u>+D</u>	<u>+B</u>	<u>+D</u>	<u>+6</u>	<u>+C</u>	<u>+6</u>
17	8	13	19	25	14	28	11	18	21
<u>-16</u>	<u>-0</u>	<u>-0</u>	<u>-16</u>	<u>-16</u>	<u>-0</u>	<u>-16</u>	<u>-0</u>	<u>-16</u>	<u>-16</u>
11	8	D	13	19	E	1C	B	12	15

مثال (29) : أوجد ناتج جمع ما يلي في النظام السادس عشر :

$$(C868)_{16} + (72D9)_{16} =$$

الحل : لايجاد المجموع نطبق قاعدة الجمع عمود عمود مع النقل إذا دعت
الضرورة الى قمة العمود التالي الواقع الى اليسار مباشرة .

		1	1	
	C	8	6	8
+	7	2	D	9
<hr/>				
المجموع في النظام العشري	19	11	20	17
التعديلات	-16	-0	-16	-16
<hr/>				
المجموع في النظام السادس عشر	1	3	B	4

و بالتالي :

$$(C868)_{16} + (72D9)_{16} = (13B41)_{16}$$

ثانياً : الطرح في النظام السادس عشر Hexadecimal Subtraction

لإيجاد الفرق السادس عشر $Y = L - M$ نوجد أولاً المتمم الخامس

عشري للعدد M عن طريق طرح كل رقم في العدد M من 15 ثم إضافة العدد واحد للحصول على المتمم لـ M ، بعد ذلك نقوم بعملية جمع متمم M إلى L .

مثال (30) : أوجد الفرق $Y = L - M$ حيث $L = 72A4$ ، $M = 4E86$.

الحل : نوجد المتمم الخامس عشري للعدد M ، حيث نطرح كل رقم في العدد

M من 15 نحصل على B179 وهو المتمم الخامس عشري للعدد M ،

نضيف العدد واحد يصبح B17A

الآن نضيف متمم M الى L نجد :

1	7	2	A	4	L
+	B	1	7	A	M متمم

الجمع العشري

التعديلات

المجموع السادس عشر

و بحذف العدد واحد الموجود على يسار الناتج نجد أن

$$Y = 72A4 - 4E86 = 241E$$

(3 - 2) المتممات العددية Complements

(3-2-1) مفهوم المتمم العددي

إن عبارة متمم تشير الى متمم الأساس ، فإذا كان لدينا العدد 6 فإن متممه العشري هو العدد 4 أي العدد الذي سوف تضيفه اليه حتى يصبح المجموع 10. تستخدم المتممات في حالتين :

- أ. في عملية تخزين الأعداد في الحاسوب ، حيث يوجد هناك العديد من الحواسيب تخزن الأعداد السالبة باستخدام متمماتها الحسابية .
- ب. في عملية الطرح ، حيث يمكن استعمال المتممات لرد عملية الطرح الى عملية جمع ، وهذا الرد مفيد بشكل خاص لتجنب تكرار عملية الاستعارة من عمود الى اخر .

هناك نوعين من المتممات في الأنظمة العددية :

1. المتمم الأساس - ناقص واحد Radix Complement-Minus One

2. المتمم الأساس Radix Complement

سوف نبحث هذين التتمين في النظام العشري وفي النظام الثنائي .

(3-2-2) المتممات العشرية في النظام العشري Decimal Complements

هناك نوعان من المتممات للأعداد العشرية ، المتمم الأول يدعى بالمتمم التساعي والمتمم الثاني يدعى بالمتمم العشري .

مثال (31): الجدول التالي يبين المتمم التساعي والعشري لبعض الأعداد في النظام العشري :

العدد العشري	751620	9672	123123	4308
المتمم التساعي	258379	0327	876876	5691
المتمم العشري	248380	0328	876877	5692

حيث حصلنا على المتم التساعي بطرح كل رقم في العدد المعطى من العدد 9 أما المتم العشري فهو يساوي الى المتمات التساعية زائد واحد .

مثال (32) : أوجد الفرق $Y = B - A$ حيث $B = 6142$ و $A = 4816$

الحل : أ. نستخدم أولا الطرح العادي :

$$\begin{array}{r} 6142 \\ - 4816 \\ \hline 1326 \end{array}$$

نلاحظ أننا يجب أن نستعير مرتين

ب. نجمع المتم التساعي لـ A المطروح وهو 5183 إلى B

$$\begin{array}{r} 6142 \\ + 5183 \\ \hline 1326 \\ \textcircled{1} \rightarrow 1 \\ \hline 1326 \end{array}$$

هنا اضفنا العدد 1 الى المجموع

جـ نضيف المتم العشري لـ A والذي هو المتم التساعي زائد واحد أي :

$$5183 + 1 = 5184$$

$$\begin{array}{r} 6142 \\ + 5183 \\ \hline 1326 \\ \textcircled{1} \end{array}$$

هنا نحمل العدد 1 للحصول على الحل .

(3-2-3) المتممات في النظام الثنائي Binary Complements

يوجد هناك متممات يستخدمان في النظام الثنائي هما المتمم الأحادي والمتمم الثنائي . إذا كان A عدداً ثنائياً ما فإن المتمم الأحادي لـ A نحصل عليه بطرح كل حد من حدود A من العدد واحد ، اما المتمم الثنائي لـ A فهو متممه الأحادي زائد واحد .

مثال (33) : الجدول التالي يبين المتمم الأحادي والثنائي لبعض الأعداد في النظام الثنائي :

العدد الثنائي	110100001111	111000111000
المتمم الأحادي	001011110000	000111000111
المتمم الثنائي	001011110001	000111001000

نلاحظ أننا نحصل على المتمم الأحادي باستبدال الصفر بواحد و الواحد بصفر في النظام الثنائي. نستطيع أن ننجز عملية الطرح في النظام الثنائي بجمع المتمم الأحادي مع العدد واحد أو بجمع المتمم الثنائي.

مثال (34) : أوجد ناتج $Y = B - A$ حيث $B = 11110000$ و $A = 10001110$

الحل : أ. ننجز عملية الطرح في النظام الثنائي العادية :

$$\begin{array}{r} 11110000 \quad B \\ - 10001110 \quad A \\ \hline \end{array}$$

$$01100010 \quad Y$$

نلاحظ أننا استعرنا الرقم واحد من الموقع الخامس .

ت. ان المتمم الأحادي لـ A هو 01110001 . نضيف متمم A الى B ثم نجمع واحد

$$\begin{array}{r}
 11110000 \quad B \\
 + 01110001 \quad A \\
 \hline
 \end{array}$$

تدعى هذه الطريقة باسم طريقة النقل الطرفي الالتفافي .

$$\begin{array}{r} 11110000 \\ + 1110010 \\ \hline \end{array}$$

نحذف الواحد نحصل على الفرق Y

حيث $A = 110011$ و $B = 101010$ ان المتمم الشائي لـ A هو 001101

$$\begin{array}{r} 101010 \\ + 001101 \\ \hline \end{array}$$

Y = 110111 نلاحظ انه لا يوجد واحد في الخانة السابقة و بالتالي

ان نطرح العدد 1000000 من العدد الناتج وبالتالي الفرق المطلوب هو :

تمارين وأسئلة عامة

- 1- حول أعداد النظام الثنائي التالية الى مكافئتها في النظام العشري :
 110.011 , 0.10101 , 0.101 , 0.11 , 10010101
- 2- حول كل من الأعداد التالية من النظام العشري الى النظام الثنائي :
 54.321 , 2468 , 0.421875 , 0.3333 , 437.40625
- 3- أوجد ناتج مايلي :
 a) $11011 + 1010 =$, b) $110.1101 + 1011.011 =$
 c) $1101 + 11100 + 1001 + 11001 =$
- 4- أوجد ناتج ما يلي :
 a) $110110 * 101 =$
 b) $111.001 * 1.11 =$
- 5- اوجد قيم عمليات الطرح التالية :
 a) $1110001 - 111011 =$
 b) $1101.0011 - 110.11011 =$
- 6- أوجد ناتج ما يلي :
 $111001 \div 1001 =$
- 7- حول الأعداد التالية الى مكافئها في النظام الثماني :
 a) $(11101010110)_2$
 b) $(1001101.01100001)_2$
- 8- حول العدد العشري $A = 1476$ الى مكافئها في النظام الثماني .
- 9- حول العدد الثماني $B = 25146_8$ الى مكافئها في النظام العشري .
- 10- حول الأعداد التالية الى مكافئها في النظام الثنائي :
 a) $(430271)_8$ b) $(21.673)_8$
- 11- حول الأعداد التالية الى مكافئها في النظام الثماني :
 a) $(11101010110)_2$, b) $(1001101.01100001)_2$

- 12- أوجد ناتج ما يلي في النظام الثماني :
- a) $3+2$, b) $6+4$, c) $7+5$, d) $2+4$, j) $5+6$
i) $7+7$, e) $3+5$
- 13- أوجد ناتج ما يلي :
- a) $(6214)_8 - (3527)_8 =$, b) $(4617263)_8 - (1423736)_8 =$
c) $(6254)_8 + (4176)_8 =$, d) $(36517)_8 + (64753)_8 =$
e) $(465.37)_8 + (31.613)_8 =$
- 14- حول الكسرين العشرين التاليين الى مكافئها في النظام الثماني :
- a) 0.4375 , b) 0.2
- 15- حول الأعداد العشرية الى مكافئها في النظام السادس عشر :
- a) 1000 , b) 15321
- 16- حول العدد السادس عشر $Z = 1A74$ الى مكافئها في النظام عشري
- 17- حول الأعداد السادس عشرة الى مكافئها في النظام الثنائي :
- a) $3D59$, b) $27.A3C$
- 18- حول الأعداد الثنائية التالية الى مكافئها في النظام السادس عشر:
- a) 10110100101110 , b) 11100.1011011011
- 19- اجمع الحدود السادس عشرة التالية :
- | | | | | | | |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | | | | | | I |
| 4 | 8 | 7 | D | 3 | 5 | E |
| <u>+3</u> | <u>+9</u> | <u>+5</u> | <u>+8</u> | <u>+B</u> | <u>+1</u> | <u>+F</u> |
- 20- أوجد ناتج ما يلي :
- a) $82C5_{16} + 9D89_{16} =$
- 21- أوجد المتمم الخمسة عشري و المتمم السادس عشر لكل من :
- a) $74B9_{16}$, b) $5C0F8_{16}$, c) $2A7600_{16}$
- 22- أوجد ناتج ما يلي :
- a) $74B64_{16} - 42AF1_{16} =$
b) $9C4D819_{16} - 23C0482_{16} =$

الفصل الرابع

تمثيل البيانات والتعليمات داخل الحاسوب

Data and Instruction Representation

(4 - 1) مفاهيم اساسية

(1-1-4) التمثيل الداخلي Internal Representation

كما ذكرنا في الفصل الثاني ، إن وحدة المعالجة المركزية (CPU) تتألف من وحدة التحكم ووحدة الحساب و المنطق ، وكل منها تحتوي على عدد من المسجلات وكل من المسجلات وظيفته الخاصة ، فمسجلات وحدة الحساب والمنطق تقوم بإستيعاب البيانات التي سيتم معالجتها ، أما مسجلات وحدة التحكم فتحوي على الأوامر و التعليمات التي سيتم تنفيذها .

إن تمثيل البيانات و الأوامر و التعليمات داخل الحاسوب يتم على شكل أرقام ثنائية 1,0 وبالتالي نحتاج لتمثيل حرف واحد الى 8 ثنائيات والتي تسمى بايت Byte ولتمثيل عدد أو جملة ذات معنى يستخدم مفهوم الكلمة Word والذي يتراوح حجمها من 8Bit الى 128 Bit ، حسب نوع الحاسوب وحدائته، وسنبين في هذا الفصل كيفية تمثيل البيانات والتعليمات داخل الحاسوب.

(4 - 1 - 2) العنوان Addressing

العنوان بالنسبة للحاسب هو ذلك الرمز الذي يشير الى أوعية التخزين داخل ذاكرة الحاسوب الرئيسية ويميزها عن بعضها البعض ، ويحدد مكان كل منها.

تنقسم الذاكرة الرئيسية الى مواقع تخزينية لكل منها عنوان يدل عليها وهذا يختلف حسب نوع الحاسوب ، حيث يمكن التمييز بين نوعين من الحواسيب:

1. حواسيب معنونة الكلمة Word Addressable Machine

في هذا النوع من الحواسيب يتم تمثيل البيانات العددية والتعليمات على شكل كلمات ، أما البيانات غير العددية كالحروف والعلامات الخاصة فيتم تمثيل كل حرف في بايت Byte وتجمع عدد من البايتات معاً لتشكيل كلمة معنونة واحدة، أي أن العنوان يكون للكلمة ، لذلك فالحواسيب معنونة الكلمة تكون أنسب للتطبيقات العلمية والرياضية ولغاتها .

2. حواسيب معنونة البايت Byte Addressable Machine

حيث تنقسم الذاكرة إلى مجموعة من البايت كل منها له عنوان أي أن العنوان هنا للبايت وليس للكلمة، يتم تمثيل البيانات التي تحتاج الى كلمة كاملة مثل البيانات العددية عن طريق تجميع عدد من البايتات متجاورة معاً والتعامل معها كوحدة واحدة عنوانها هو عنوان البايت الأولى، هذا النوع من الحواسيب يكون أنسب للتطبيقات التي لا تحتاج عمليات حسابية كثيرة وكذلك التطبيقات التي تعتمد على البيانات غير العددية .

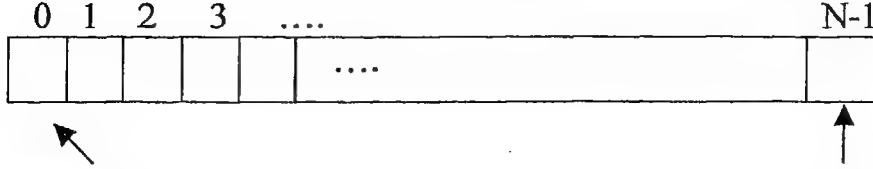
(4 - 2) تمثيل الأعداد الصحيحة :

Representation Integer Number:

المقصود بالأعداد الصحيحة، تلك القيم العددية التي لا تحتوي على فاصلة عشرية (نقطة كسرية)، هذه العلامة الكسرية يفترض وجودها دائماً في موقع ثابت في أقصى اليمين من العدد بدون ان تظهر، فالعدد يكون صحيحاً فقط إذا خلا من النقطة الكسرية الظاهرة ، هناك أكثر من طريقة لتمثيل القيم العددية الصحيحة داخل الحاسوب .

(1-2-4) تمثيل العدد وإشارته :

وفقا لهذه الطريقة يتم تمثيل القيم الصحيحة داخل الحاسوب في مسجل ينقسم الى N من المواقع الثنائية مرقمة من (0) الى $(N - 1)$.



نلاحظ من الشكل السابق انه يُخصص اقصى موقع ثنائي في اليسار لتمثيل الإشارة فيوضع العدد (0) لتمثيل الإشارة الموجبة $(+)$ أو العدد (1) لتمثيل الإشارة السالبة $(-)$ ، أما بقية مواقع المسجل فيخصص لتمثيل العدد الصحيح وذلك من اليمين الى اليسار ، العلامة الكسرية يفترض وجودها في اقصى اليمين ولا تأخذ موقعاً.

مثال (1) : الجدول التالي يعرض لنا امثلة ايضاحية لكيفية تمثيل الأعداد الصحيحة سالبة أو موجبة وذلك في مسجل يتألف من ثمانية ثنائيات أي 8Bit .

مثال	العدد الثنائي	تمثيله في مسجل طوله 8 bit
1	110110	0 0 1 1 0 1 1 0
2	- 10110	1 0 0 1 0 1 1 0
3	10100	0 0 0 1 0 1 0 0
4	- 100	1 0 0 0 0 1 0 0
5	- 1111010	1 1 1 1 1 0 1 0
6	111100101	0 1 1 0 0 1 0 1

من الأمثلة السابقة نستنتج ما يلي :

1. يُخصص اقصى موقع من اليسار لتمثيل الإشارة .
2. العلامة الكسرية يفترض وجودها في أقصى اليمين و لا تأخذ موقعاً .
3. تمثيل العدد الصحيح يكون من اليمين الى اليسار .

4. اذا كان عدد مواقع العدد أقل من عدد مواقع المسجل ، تملأ بقية مواقع المسجل بالأصفار دون التعرض لموقع قيمة الإشارة .

5. اذا كان عدد مواقع العدد اكبر من عدد مواقع المسجل، فان اعداد اليسار التي تزيد عن حجم المسجل تحذف دون التعرض لموقع وقيمة الإشارة ، ففي المثال (1) البند السادس يتم تمثيل السبعة أرقام الأولى من اليمين إلى اليسار ويتم حذف الرقمين الأخيرين (11) مع ملاحظة أن لهما اكبر قيمة في العدد .

(2-2-4) التمثيل بالمتمم الثنائي :

في هذه الطريقة تمثل الإشارة ومتمم العدد، سواء كان ذلك للقيم الموجبة أو السالبة :

1. تمثيل القيم الموجبة بالمتمم الثنائي : متمم القيمة الموجبة هو القيمة نفسها . فلو أردنا تمثيل القيمة 1001101 في مسجل ذو ثمانية مواقع فان ذلك يكون على الشكل التالي :

0	1	0	0	1	1	0	1
---	---	---	---	---	---	---	---

2. تمثيل القيم السالبة : ليكن لدينا القيمة (51 -) ولنرى خطوات تمثيل هذه القيمة :

-110011	:	القيمة (51 -) تعادل بالنظام الثنائي
- 0110011	:	باستخدام مسجل ذو ثمانية مواقع يكون
11001101	:	و بالحصول على المتمم الثنائي تصبح

أي تمثل على الشكل التالي :

1	1	0	0	1	1	0	1
---	---	---	---	---	---	---	---

(3-2-4) حدود التمثيل ومشكلة الفائض :

يتم تخزين (تمثيل) القيم الصحيحة ضمن حدود معينة تتوقف على طول المسجل وذلك كالآتي :

أ. اكبر قيمة صحيحة موجبة يمكن تمثيلها وفق نظام النقطة الثابتة نحصل عليها من العلاقة $(2^{n-1} - 1)$ حيث تمثل N عدد المواقع الثنائية في المسجل .

فاذا كان لدينا مسجل بطول ثمانية مواقع ثنائية فان اكبر قيمة يمكن تخزينها في المسجل هي :

$$(2^{8-1} - 1) = 2^7 - 1 = 127$$

وهذه القيمة تمثل كالتالي :

0	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

أي اننا لا نستطيع تمثيل القيمة $128 +$ أو أي قيمة اكبر منها في هذا المسجل يحوي 8 مواقع (8Bit) .

ب. اقل قيمة صحيحة يمكن تمثيلها في المسجل السابق هي :

$$(-2^{8-1} + 1) = -2^7 + 1 = -128 + 1 = -127$$

أي اننا لا نستطيع تمثيل القيمة $128 -$ أو اقل منها في هذا المسجل ، وبالتالي نحصل على تمثيل خاطئ عندما نريد تمثيل قيمة اكبر أو اصغر من المجال (, $127 +$, -127) .

من الجدير بالذكر أن الحاسبات الحديثة تستخدم مسجلات بسعة 32 موقع ثنائي ومثل هذه المسجلات تكون قادرة على استيعاب قيم حتى 2147483647 أي $(2^{32-1} - 1)$ وبالتالي يكفي أن نعرف طول المسجل في الحاسوب لمعرفة العدد الممكن تخزينه .

(4 - 3) تمثيل الأعداد ذات النقطة العائمة

Floating Point Representation

و هي اكثر طرق تمثيل الأعداد أهمية وذلك لأنها تتميز بما يلي :

1. امكانية تمثيل الأعداد الكبيرة والصغيرة التي لا يمكن تمثيلها باستخدام طريقة النقطة الثابتة .

2. الحصول على درجة دقة عالية جداً بالمقارنة بأسلوب النقطة الثابتة خاصة بالنسبة للجزء الكسري .

3. تعتبر أنسب الطرق لتمثيل الأعداد الكسرية أو التي تحتوي على جزء صحيح وآخر كسري .

4. يمكن زيادة درجة الدقة في التمثيل عن طريق ما يسمى بالدقة المضاعفة .

5. يمكن بواسطتها تمثيل الأعداد المركبة بجزئها الحقيقي والتخيلي .

لإيضاح فكرة الأعداد ذات النقطة العائمة وتمثيلها ، نفترض ان لدينا عددين الأول هو 1200000000 والثاني هو 0.000000009 فإننا لا نستطيع تمثيلها داخل الحاسوب باستخدام أسلوب النقطة الثابتة ، وذلك لكبر الأول وصغر الثاني ، ولكننا اذا أعدنا صياغة العددين بصورة كسرية أسيه يصبحان كالتالي :

$$12000\ 000\ 000 = 0.12 \times 10^{11}$$
$$0.000\ 000\ 00\ 9 = 0.9 \times 10^{-8}$$

أي اننا نحصل على صيغة رقمية ممكنة التمثيل . هذه الصيغة العددية هي التي تسمى بشكل الأعداد عائمة النقطة وصيغتها العامة تكون على الشكل التالي :

$$\pm M * R^E$$

حيث ان :

M - تعبر عن الصيغة الكسرية للعدد و تسمى القاعدة .

R - تعبر عن اساس النظام المستخدم

E - تعبر عن الأس المرفوع اليه الاساس

وهذا يعني انه لتمثيل أي رقم وفق نظام النقطة العائمة ، ينبغي أولاً وصفه في الصيغة السابقة، والتي تسمى بالشكل المعياري و من ثم تخزينه في مسجل ذي مواصفات تتفق وهذه الصيغة.

(1-3-4) الشكل المعياري Standard Form

عند تمثيل العدد باستخدام صيغة الأعداد عائمة النقطة يجب الأخذ بعين الاعتبار ما يلي :

1. العلامة الكسرية تكون دائماً في أقصى اليسار من العدد مثل القاعدة للعدد العشري 0.1275 أو القاعدة للعدد الثنائية 0.101011 .
2. من الخطأ ان تكون القاعدة مثلاً على الشكل التالي 0.000234 والصحيح يجب ألا يكون هناك اصفار على يمين العلامة الكسرية ، وإن وجد يجب تحويل العدد إلى صيغة تحذف بها الأصفار ، أي الشكل :

$$0.234 * 10^{-3}$$

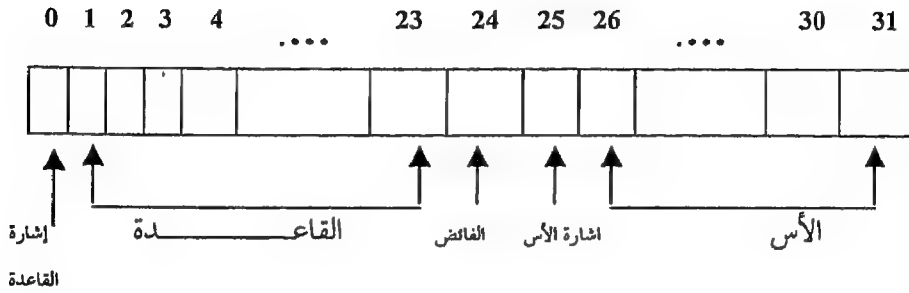
مثال (2) : الجدول التالي يبين بعض الامثلة لتحويل الأعداد من الصيغة المستعارة إلى الصيغة المعيارية .

العدد	الشكل المعياري
4234	$0.4234 * 10^4$
123.42	$0.12342 * 10^3$
- 1.234	$- 0.1234 * 10^1$
0.00051	$0.51 * 10^{-3}$
- 0.050	$- 0.5 * 10^{-1}$
1101.01	$0.110101 * 2^{+4}$
10.1101	$0.1011 * 2^2$
- 0.1011	$- 0.1011 * 2^0$
0.0000011	$0.11 * 2^{-5}$

(2-3-4) المسجل ذو النقطة العائمة

Floating Point Register

يتم تمثيل الأعداد بعد تحويلها إلى الشكل المعياري في مسجل خاص ويختلف هذا المسجل من نظام إلى نظام آخر سواء في طوله أو في شكله ، على سبيل المثال إذا كان طول المسجل هو 32 بت فإنه يمكن تصور الشكل التالي للمسجل ذي النقطة العائمة :



حيث أن :

1. يتم تخصيص موقع واحد لإشارة القاعدة وآخر لإشارة الأس .
2. يتم تخصيص عدة مواقع (23 موقعا هنا) لتمثيل القاعدة .
3. تخصص عدة مواقع لتمثيل الأس .
4. يُخصص موقع واحد للفائض حيث يحتوي على العدد 1 إذا كانت قيمة الأس أكبر من المواقع المخصصة لتخزينه، وتحتوي على القيمة 0 إذا كانت قيمة الأس في حدود تلك المواقع .

مثال (3) : ليكن لدينا مسجلا ذو 12 موقعا ثنائي، ستة منها للقاعدة وثلاثة للأس وموقع واحد لكل من إشارة القاعدة وإشارة الأس والفائض .

1- كيف يتم تمثيل العدد العشري 7.5 .

الحل : لتمثيل العدد 7.5 نتبع الخطوات التالية :

أ. نحصل على المكافئ الثنائي للرقم و هو 111.1

ب. نحوله الى الشكل المعياري فيكون $0.1111 * 2^3$

ج — نمثل الصيغة المعيارية في المسجل ذي النقطة العائمة كالتالي :

0	1	1	1	1	0	0	0	0	0	1	1
الأس		إشارة		الفائض		القاعدة		إشارة		القاعدة	
الأس		إشارة		الفائض		القاعدة		إشارة		القاعدة	

2 - كيف يتم تمثيل العدد 11.625

الحل : العدد الثنائي المكافئ هو 1011.101 ، نحول هذا العدد الى الشكل

المعياري فيكون $0.1011101 * 2^4$ و بالتالي يمثل على الشكل التالي :

0	1	0	1	1	1	0	0	0	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---

نلاحظ ضياع آخر عدد من يمين قاعدة العدد وذلك بسبب عدم كفاية الجزء المخصص للقاعدة

3- العدد العشري 131.5 يناظرها العدد 10000011.1 شكلة المعياري

$0.100000111 * 2^8$ وبالتالي يتم تمثيله على الشكل التالي:

0	1	0	0	0	0	0	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---

نلاحظ ان قيمة الأس اكبر من سعة الجزء المخصص له وبالتالي ظهر العدد 1 في موقع الفائض . كما نلاحظ أنه تم اهمال 111 من اليمين في القاعدة لعدم اتساع الجزء المخصص لها .

4- القيمة الثنائية 0.000001 يعبر عنها معيارياً $0.1 * 2^{-5}$ و يمثل داخل المسجل

كما يلي :

0	1	0	0	0	0	0	0	1	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---

نلاحظ أن الأس السالب قد تم تمثيله في صورة متممة ثنائي .

ملاحظة : لتمثيل الأعداد بدقة مضاعفة نستخدم مسجلين ، كما يمكن تخزين القيم المركبة داخل مسجلين بحيث يخزن الجزء الحقيقي لهذه القيم داخل مسجل والجزء التخيلي داخل المسجل الآخر .

(4 - 4) تمثيل الأحرف والرموز

Character Representation

المقصود بالأحرف و الرموز ما يلي :

- الأعداد التي لا تدخل في العمليات الحسابية (0 , 1 , ... , 9) والتي تعامل كرموز وليس كأعداد .
 - الحروف الهجائية (A , ... , Z) و (a , ... , z) .
 - الرموز والعلامات الخاصة () , ... , () , " , ; , # , \ , * , ...) .
- يتم تمثيل هذه الأحرف والرموز في وحدة تخزينية تسمى بايت والبايت كما ذكرنا تتكون من ثمانية مواقع ثنائية . على الرغم من تنوع الأنظمة المستخدمة في تمثيل الحروف إلا أنها تعتمد جميعها على تقسيم البايت إلى جزئين .
- أ. قسم النطاق Zone Area : ويتكون من أربعة مواقع ثنائية أو من موقعين ثنائيين حسب النظام المستخدم .
- ب. القسم العددي Numeric Area ويتكون من أربعة مواقع ثنائية .
- يوجد هناك أنظمة لتمثيل الحروف و الرموز أشهرها :

(1-4-4) نظام (Binary Coded Decimal (BCD)

استخدمت هذه الشفرة في الأجيال المبكرة للحواسيب ويندر استخدامها في الوقت الحالي ، حيث تمثل الحروف والرموز والأعداد في بايت ذات سنت مواقع، والبايت في هذا النظام يستوعب عددين ثنائيين (العدد الثماني يمثل في ثلاثة مواقع)، حيث قسم النطاق يتكون من موقعين ، انظر الجدول (1) .

مثال (4) : تمثل كلمة (Ali) في ثلاث بايتات كالآتي :

A					L							I					
1	1	0	0	0	1	1	0	0	0	1	1	1	1	1	0	0	1

(2-4-4) نظام EBCDIC

Extended Binary Coded Decimal Interchange Code

يعتمد هذا النظام على البايت ذات الثمانية مواقع، أربعة للنطاق و أربعة للقطاع العددي، والبايت يستوعب عددين ست عشريين أنظر الجدول (1)
مثال : تمثل كلمة (TO) في نظام EBCDIC كالآتي :

T								O							
1	0	1	1	0	1	0	0	1	0	1	0	1	1	1	0

(3-4-4) نظام (ASCII)

American Standard Code for Information Interchange

يعتمد هذا النظام على البايت ذات ثمانية مواقع ، أربعة للنطاق وأربعة للقطاع العددي ، انظر الجدول (1) .
مثال : تمثل الكلمة (TO) في نظام ASCII كالآتي :

T								O							
1	0	1	1	0	1	0	0	1	0	1	0	1	1	1	1

الجدول (1)

الحرف	BCD	EBCDIC	ASCII
A	11 0001	1100 0001	1010 0001
B	11 0010	1100 0010	1010 0010
C	11 0011	1100 0011	1010 0011
D	11 0100	1100 0100	1010 0100
E	11 0101	1100 0101	1010 0101
F	11 0110	1100 0110	1010 0110
G	11 0111	1100 0111	1010 0111
H	11 1000	1100 1000	1010 1000
I	11 1001	1100 1001	1010 1001
J	10 0001	1101 0001	1010 1010
K	10 0010	1101 0010	1010 1011
L	10 0011	1101 0011	1010 1100
M	10 0100	1101 0100	1010 1101
N	10 0101	1101 0101	1010 1110
O	10 0110	1101 0110	1010 1111
P	10 0111	1101 0111	1011 0000
Q	10 1000	1101 1000	1011 0001
R	10 1001	1101 1001	1011 0010
S	01 0010	1110 0010	1011 0011
T	01 0011	1110 0011	1011 0100
U	01 0100	1110 0100	1011 0101
V	01 0101	1110 0101	1011 0110
W	01 0110	1110 0110	1011 0111
X	01 0111	1110 0111	1011 1000
Y	01 1000	1110 1000	1011 1001
Z	01 1001	1110 1001	1011 1010
0	00 1010	1111 0000	0101 0000
1	00 0001	1111 0001	0101 0001
2	00 0010	1111 0010	0101 0010
3	00 0011	1111 0011	0101 0011
4	00 0100	1111 0100	0101 0100
5	00 0101	1111 0101	0101 0101
6	00 0110	1111 0110	0101 0110
7	00 0111	1111 0111	0101 0111
8	00 1000	1111 1000	0101 1000
9	00 1001	1111 1001	0101 1001

(4 - 5) تمثيل الأوامر والتعليمات

من اجل التعرف على كيفية تمثيل الأوامر والتعليمات يجب التعرف على مفاهيم التعليمات أو الأمر والبرنامج ثم ننتقل الى الكيفية التي يتم بها تمثيلها في ذاكرة الحاسوب .

التعليمات Instruction أو الأمر Order هو ايعاز للحاسوب لانجاز عملية معينة مثلاً :

التعليمات $Z=A+B$ إيعاز للحاسوب بجمع قيمة A الى B وتخزينها في Z

التعليمات Read (x) إيعاز للحاسوب بإدخال قيمة X للذاكرة .

التعليمات Write (y) إيعاز للحاسوب بإخراج قيمة y كنتائج مطبوعة .

كل مجموعة مترابطة منطقياً من التعليمات والأوامر والمتعلقة بمعالجة مشكلة محددة تسمى برنامجاً Program .

إن عملية معالجة البرنامج تتطلب أن يكون البرنامج مخزناً في الذاكرة ويتم التعامل معه تعليمات ، تعليمات بالتسلسل داخل وحدة المعالجة المركزية .
يحتل الأمر الواحد في ذاكرة الحاسوب كلمة واحدة ، وهذا الأمر أو التعليمات ينقسم الى مقطعين :

1. مقطع رمز العملية (Operation Code) والخاص بتوريد العملية التي يراد من الحاسوب انجازها مثل ADD أو LOAD وبعبارة أخرى هو رمز فعل الأمر في هذه التعليمات .

2. مقطع العنوان (Address Section) والخاص بتحديد عنوان أو عناوين الوحدات التخزينية التي ستخضع لعملية المعالجة ، وقد يسمى Operand Field .

فعلى سبيل المثال اذا كان لدينا الأمر التالي بلغة الحاسوب Assembly
Add 542

والذي يعني جمع محتويات المخزن الذي عنوانه 542 تمثل عنوان المخزن الذي يحتوي على البيانات المراد جمعها الى المراكم ، وبالتالي فان الكلمة أو المسجل الذي سيستوعب هذا الأمر سيكون كالتالي :

ADD	542
-----	-----

لنفترض ان طول الكلمة التي تستوعب الأمر هو (16 bit) منها اربعة لرمز العملية والباقي للعنوان ، ولنفترض ايضا ان المكافئ الثنائي لعملية الجمع هو 0101 فإننا نستطيع تمثيل الأمر السابق في الكلمة ذات الستة عشر موقعا ثنائيا كالتالي :

0	1	0	1	0	0	1	0	0	0	0	1	1	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

مع ملاحظة ان : $(542)_{10} = (1011011100)_2$ وليس هذا هو الشكل الوحيد لتمثيل الأوامر إذ أنه يمكن التمييز بين الاشكال الثلاثة التالية:

أ. الأمر أحادي العنوان وقد تم عرضه اعلاه .

ب. الأمر ثنائي العنوان ، حيث يكون هناك عنوانان في الأمر الواحد ، فمثلا :

ADD X , y

والذي يعني جمع القيمة الموجودة في المخزن الذي عنوانه X الى القيمة الموجودة في المخزن الذي عنوانه Y وتخزين الناتج في المخزن الثاني :

X								Y							
0	1	0	1	1	1	0	1	1	1	0	0	1	1	0	1

على اعتبار ان عنوان المخزن الأول والذي تمثله X هو 55 و عنوان المخزن الثاني والذي تمثله Y هو 13 .

ج — الأمر ثلاثي العنوان ، يحتوي الأمر على ثلاثة عناوين ويتم تمثيله بنفس الأسلوب السابق .

فمثلاً الأمر :

$$Z = X + Y$$

و الذي يعني جمع قيمة X و Y وتخزينها في المخزن الذي عنوانه Z لو اعتبرنا أن عنوان المخزن الأول 33 و عنوان المخزن Y هو 3 و عنوان المخزن Z هو 36 فيكون تمثيله على الشكل التالي :

X										Y					Z						
0	1	0	1	1	0	0	0	0	1	0	0	0	0	1	1	1	0	0	1	0	0

أسئلة وتمارين عامة

1. كيف يتم تمثيل الأعداد الثنائية التالية في مسجل طوله 8 بت ؟
أ. 100110011 ب. 10110 ج. 10100111
2. كيف يتم تمثيل الأعداد الثنائية التالية في مسجل طوله 12 بت
أ. 462 - ب. 1357 ج. 63 -
3. استخدام التمثيل بالمتعم الثنائي للأعداد التالية في مسجل ذو ثمانية مواقع :
أ. 101010101 ب. 111000 ج. 98
د. 112 - هـ. 135 و. 111111
4. اكتب الشكل المعياري للأعداد التالية :
أ. 65321 ب. 632.145 ج. 22.34 -
د. 0.00101 هـ. 0.000111 - و. 11011.1011
5. ليكن لدينا مسجلا ذو 12 موقعا ثنائيا، ستة منها للقاعدة ، كيف يتم تمثيل
الأعداد التالية :
أ. 18.5 - ب. 132.5 ج. 6 -
د. 0.100001 هـ. 1101.0011 و. 1111.10011 -

الفصل الخامس

الكيان المنطقي للحاسوب (البرمجيات)

SOFTWARE

(5 - 1) ماهية البرمجيات :

من المعروف أن نظام الحاسوب يتكون من جزأين رئيسيين :

1. المكونات المادية Hardware والتي سبق شرحها في الفصل الثاني .

2. برمجيات الحاسوب Software .

و البرنامج هو عبارة عن مجموعة من التعليمات تكتب بلغة معينة وتخزن في الذاكر لتخبر الحاسوب ماذا وكيف يعمل لتنفيذ مهمة معينة وتمكنه من إدارة ومراقبة وتنظيم مكوناته المادية لتحقيق تلك المهمة ، بعبارة أخرى برامج الحاسوب بمثابة الروح من الجسد وإن أي نظام حاسوبي بدونها يصبح عديم الجدوى.

(5 - 2) أنواع برمجيات الحاسوب

هناك نوعان رئيسيان من برمجيات الحاسوب يطلق على إحداها برمجيات

النظام System Software والنوع الآخر برمجيات التطبيقات Application Software.

(5-2-1) برمجيات النظام : System Software

وهي البرامج المعدة بواسطة الجهة الصانعة للحواسيب والتي يستخدمها الحاسوب ليتحكم ويوجه ويشرف على نظام الحاسوب بأكمله من مكوناته المادية وبرامج التطبيقات ، وهذه البرامج إما أن تكون مخزنة تخزيناً دائماً في ذاكرة القراءة

فقط ROM وبعضها الآخر يكون مخزناً على وسيط خارجي (أقراص ممغنطة) تستدعى وقت الحاجة إلى الذاكرة المؤقتة RAM .

ويمكن تقسيم برمجيات النظام إلى ثلاثة مجموعات أساسية هي :

أولاً : أنظمة التشغيل Operating Systems

وهي الأنظمة والبرامج المسؤولة عن عمليات تشغيل الأجهزة والمعدات وفق نظام خاص وخطوات مرتبة من أجل معالجة البيانات والإشراف على تشغيل جميع برامج التطبيقات ، وهناك العديد من أنظمة التشغيل المستخدمة والمتاحة والتي يتم أعدادها بواسطة منتجي الحواسيب ونظام التشغيل MS-DOS يعتبر أكثر نظم التشغيل شهرة للتعامل مع الحواسيب الصغيرة ، ونظام التشغيل Unix يعتبر من أنظمة التشغيل الشهيرة للحواسيب المتوسطة .. (وستناول موضوع أنظمة التشغيل بالتفصيل لاحقاً إن شاء الله) .

ثانياً : مترجمات اللغات Language Compilers

وهي البرامج المسؤولة عن تحويل أوامر وإيعازات البرنامج الى إشارات يستطيع الحاسوب تفسيرها والعمل على أساسها ، أي إلى لغة الآلة والتي تتألف بدورها من تكوينات من النظام الثنائي الصفر والواحد ، ويطلق على البرنامج المكتوب بلغة عالية المستوى بالبرنامج المصدر أو (رمز المصدر Source Code)، ويطلق على البرنامج المكتوب بلغة الآلة البرنامج الهدف أو (رمز الهدف Object Code) .

وهناك نوع آخر من برامج الترجمة يطلق عليها اسم المفسر Interpreter والفرق بين المترجم والمفسر هو أن المترجم أسرع من المفسر بكثير وذلك لأن المترجم يترجم البرنامج المصدر مرة واحدة ثم يقوم بتخزين برنامج الهدف برمز الآلة كملف Object Code File وذلك لاستخدامه عند الحاجة دون أن يترجم

البرنامج المصدر مرة أخرى ، أما المفسر فإنه يقوم بترجمة البرنامج المصدر خطوة خطوة أثناء تنفيذ البرنامج ولذلك فإن البرنامج المترجم يحتاج حيز في الذاكرة أكبر من ذلك الذي يحتاجه البرنامج المفسر ، كما أن البرنامج المفسر يمكن إيقافه أثناء التشغيل عند أي موضع ثم إعادة تشغيله مرة أخرى وذلك بعكس البرنامج المترجم.

ثالثاً : البرنامج المساعد Utilities

وهو عبارة عن مجموعة من البرامج المنفصلة التي تؤدي كل منها وظيفة أو وظائف محددة ، تستخدم أساساً للسيطرة والتحكم في مكونات الحاسوب المادية وغير المادية بالإضافة إلى تنفيذ وظائف نظام التشغيل بشكل أسهل ، ومن أمثلة هذه البرامج أدوات الحاسوب الشخصي PC-Tolls وبرامج أدوات نورتن Norton Utilities .

(2-2-5) برمجيات التطبيقات Application Software

وهي تلك البرامج التي تعد من قبل مستخدمي الحاسوب من المبرمجين بهدف حل مشكلة معينة أو مسألة معينة تواجههم ، وذلك فإن هذا النوع من البرامج تشتمل تقريباً جميع أنشطة الحياة ، ويمكن لمستخدمي الحاسوب الحصول على برامج التطبيقات عن طريقين :

1. أن يقوم المستخدم بكتابة البرامج بنفسه في العادة يستخدم لذلك إحدى اللغات العالية المستوى (وستتناول ذلك في الفصلين السابع والثامن) .
2. أن يقوم المستخدم بشراء برامج جاهزة خططت و كتبت بواسطة مجموعة من المبرمجين المتخصصين ، ويوجد حالياً الآلاف من هذه البرامج وفي جميع المجالات المختلفة ، وهنا لا بد من الإشارة إلى أن برامج التطبيقات تكتب لنظام تشغيل معين أي أن برامج التطبيقات المكتوبة لنظام تشغيل محدد لا يمكن تنفيذها باستخدام نظام تشغيل آخر بدون إجراء تعديلات ، وحتى إذا كانت الحواسيب المختلفة تشترك في نفس نظام التشغيل ولكنها مصنعة بواسطة

منتجين مختلفين فانه لا بد من إجراء بعض التعديلات على البرامج ، ولذلك فمن المهم للمستخدم قبل شراء أي حزمة من البرامج التطبيقية معرفة نوع الحاسوب الذي يعمل عليه ونوع نظام التشغيل المستخدم .

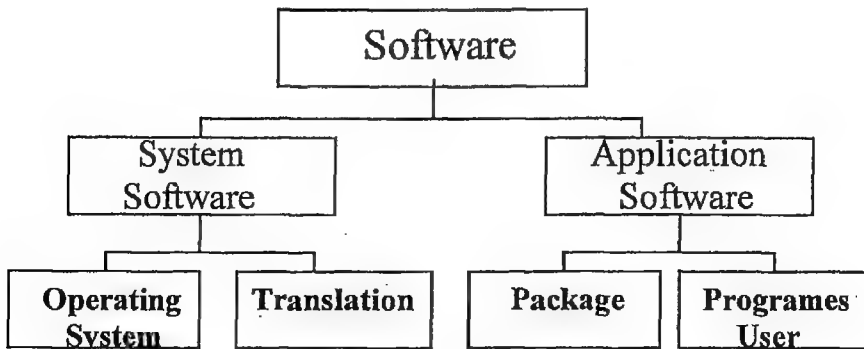
وبشكل عام تقسم برمجيات التطبيقات الى نوعين :

أولاً : برمجيات التطبيقات الجاهزة :

وتدعى أيضاً الحزم البرمجية (Packages) ومن أمثلة تلك البرامج برامج معالجة النصوص Word Processors وبرامج الجداول الإلكترونية Spread Sheets ، وبرامج نظم ادارة قواعد البيانات Data Base Management Systems ، وغيرها ... ، وهذه النوعية من البرامج يكتبها مبرمجون متخصصون وسنلقي الضوء عليها في الفقرات اللاحقة ان شاء الله.

ثانياً : برمجيات التطبيقات الخاصة :

وهي برامج لحل مسائل أو مشاكل خاصة يعدها مستخدموا الحاسوب وكل مستخدم في مجاله مثل البرامج الخاصة لحل المسائل الرياضية المحددة وبرامج حل مسائل النقل والبرمجة الخطية وبرامج تحليل البيانات الإحصائية وإجراء الاختبارات الإحصائية، وبرامج الأعمال خاصة بالشركات والهيئات ... والشكل (1) يلخص الأقسام المختلفة لبرمجيات الحاسوب .



الشكل (1)

(3 - 5) نظام التشغيل Operating System

(1-3-5) تعريف نظام التشغيل

إن التقدم العلمي والتكنولوجي الذي يشهده العالم اليوم في إنتاج الحواسيب الإلكترونية وتطبيقاتها الواسعة في شتى الميادين ، أدى إلى وجوب معرفة ودراسة نظم التشغيل المختلفة التي تساعد المستخدم على التعامل مع الأنواع المختلفة من الحواسيب ، حيث أن اختيار الحاسوب المناسب لا يعتمد فقط على المكونات المادية Hardware وعلى اللغات العالية المستوى والبرامج التطبيقية التي تتعامل مع هذه المكونات بل يعتمد بصورة أساسية على نظم التشغيل المستخدمة والتي تتعامل مع تلك المكونات .

نظام التشغيل يمثل حلقة الاتصال بين المستخدم والحاسوب ، فالمستخدم يتعامل مع مكونات الحاسوب المختلفة من خلال نظام التشغيل ، حيث أنه ينظم العمليات التي يقوم بها الحاسوب .

ويمكن تشبيه نظام التشغيل بالمدير الذي يدير مشروعاً معيناً ولكي يكتب لهذا المشروع النجاح وبكفاءة عالية فإن على المدير أن يقوم بتنظيم أنشطة المشروع والموارد المستخدمة في تنفيذ هذه الأنشطة بهدف تنفيذ المشروع في أقل وقت ممكن وبأكفاً استغلال للموارد المتاحة . حيث أن نظام التشغيل يقوم بنفس العملية ، مع الأخذ بعين الاعتبار أن الموارد في هذه الحالة هي مكونات الحاسوب والتي تشمل الوحدات التالية :

وحدة المعالجة المركزية	Central Processing Unit
الذاكرة الرئيسية	Main Memory
وحدات الإدخال والإخراج	Input/Output Units

كما أن هذه الموارد تشتمل أيضاً على الموارد البشرية المتمثلة بمشغل الحاسوب والمبرمج ومحلل النظم .. الخ
أما الأنشطة فهي جميع العمليات التي يقوم بها الحاسوب من تحميل للبرامج وإدخال للبيانات ومعالجة لهذه البيانات .. الخ
من هذه المقدمة البسيطة يتضح أن نظام التشغيل يمثل العمود الفقري لنظام الحاسوب والذي بدونونه يصعب تعامل الإنسان مع الحاسوب والاستفادة من إمكانياته .

(2-3-5) الخصائص العامة لنظام التشغيل :

في الوقت الحاضر تتنافس نظم التشغيل فيما بينها للوصول إلى الصورة المثالية لاستخدام الحاسوب ومكوناته ، ونظام التشغيل المثالي يتمتع بالخصائص التالية :

1. التحكم والسيطرة على مكونات الحاسوب وذلك بالربط بين وحدة التشغيل المركزية التي تعتبر بمثابة عقل الحاسوب وبين باقي المكونات .
2. أداء العمليات الأساسية التي تساعد المستخدم على التعامل مع مكونات الحاسوب مثل نسخ الملفات ومسح ونسخ الأقراص وتشكيلها .. الخ وذلك باستخدام مجموعة من الأوامر Commands يسهل على المستخدم حفظها، والتعامل مع الحاسوب من خلالها.
3. التعامل مع برامج التطبيقات Application Programs مثل الجداول الإلكترونية وبرامج معالجة الكلمات ونظم إدارة قواعد البيانات .
4. تنظيم الأعمال التي يقوم بها الحاسوب Job Control .
5. القدرة على أداء عدة وظائف في نفس الوقت Multitasking .
6. السماح لعدد من المستخدمين بالتعامل مع الحاسوب في وقت واحد Multiusers .

7. القدرة على التعامل مع عدد كبير من أجهزة الحواسيب المصنعة في شركات مختلفة Portability .

8. القدرة على التعامل مع شبكات الحاسوب التي تستخدم وحدات طرفية بعيدة Remote Terminals .

9. القدرة على إضافة إمكانيات جديدة الى نظام التشغيل أو إلغاء إمكانيات أصبحت غير مطلوبة وذلك دون الحاجة إلى كتابة برنامج جديد لنظام التشغيل.

(3-3-5) أنواع نظم التشغيل :

يمكن تقسيم نظم التشغيل إلى نوعين أساسيين وعلى الشكل التالي :
أولاً : نظم التشغيل حسب نوع أجهزة التخزين المساعدة والتي يمكن أن تنقسم إلى فرعين هما :

1. نظام التشغيل القرصي Disk Operating system واختصاراً DOS ، والقرص على نوعين إما صلب (Hard) أو مرن (Floppy) .

2. نظام التشغيل الشريطي Tape Operating System واختصاراً TOS والذي يتم تخزين المعلومات فيه على الشريط المغنط.

ثانياً : نظم التشغيل حسب العلاقة بين المستخدم والحاسوب وعدد المستخدمين الذين يعملون في وقت واحد على الحاسوب ، وهذه النظم يمكن تقسيمها إلى أربعة أنواع وهي:

1. نظم تشغيل البرنامج الواحد Single Program System
في هذا النظام تعمل جميع مكونات الحاسوب على تنفيذ برنامج واحد فقط في الوقت الواحد ، والمستخدم يتعامل بمفرده مع الحاسوب ويراقب بنفسه أجهزة الإدخال والإخراج ، وهذا النوع متوفر في جميع الحواسيب الشخصية .

2. نظام البرمجة المتعددة Multitasking System

وهذا النوع من أنظمة التشغيل يسمح بتخزين أكثر من برنامج واحد في الذاكرة الرئيسية وفي نفس الوقت تكون جميع مكونات الحاسوب في حالة تشغيل لعدد من البرامج في آن واحد بحيث يكون أحد البرامج في حالة قراءة وآخر في حالة معالجة ، وآخر في حالة طباعة ... والبرامج المراد معالجتها والتي تنتظر دخولها إلى الحاسوب ، تكون مرتبة بشكل طابور على جهاز تخزين مساعد وإن عدد البرامج التي يمكن تخزينها في الذاكرة في وقت واحد يعتمد على سعة الذاكرة ونوع نظام التشغيل وحجم البرامج المطلوب تنفيذها. لذلك تتطلب مثل هذه الأنظمة عدة أمور منها :

- 1- توفر ذاكرة رئيسية ذات سعة كبيرة .
- 2- توفر أجهزة تخزين مساعدة سريعة الأداء .
- 3- توفر وحدة الحساب والمنطق من النوع السريع جدا .

3. نظام المشاركة الزمنية Time Sharing

في هذا النوع من النظم يتم تقسيم وقت المعالجة بين مجموعة من البرامج في الذاكرة الرئيسية بحيث يتم إعطاء الفرصة لجميع البرامج في الحصول على نصيب صغير من عمل وحدة المعالجة المركزية بالتساوي و بالتناوب ، ثم ظهر فيما بعد نظام التشغيل التفاعلي المتعدد البرمجة ، والذي يسمح لعدد من المستخدمين بالتفاعل مباشرة مع جهاز الحاسوب عن طريق تنفيذ برامجهم من وحدات طرفية متعددة متصلة بالجهاز الرئيسي .

أما أهم ميزات هذا النظام :

- سهولة الاتصال المباشر بين المستخدم وجهاز الحاسوب الرئيسي .
- إتاحة الفرصة لعدد كبير من المستخدمين بالتعامل مع الحاسوب بآن واحد.
- انخفاض التكاليف .

4. نظام المشاركة الزمنية الحقيقية Real Time System
وهذا النظام هو تطوير للنظام السابق إذ أن المستخدم لا ينتظر وقتاً معيناً لتنفيذ برنامجه بل يكون التجارب لحظي وفوري مهما يكن عدد المستخدمين على الأجهزة الطرفية .
من أمثلتها نظم حجز أماكن الطيران ، نظم المراقبة الجمركية على نقاط الحدود..

5. نظام الذاكرة الافتراضية Virtual Memory
يستخدم في الدواكر التي سعتها أكبر من السعة الحقيقية لذاكرة الكمبيوتر ، ويعتمد هذا النظام على تقسيم البرنامج إلى مقاطع وتنفيذ المقطع تلو المقطع .
(5 - 3 - 4) أنواع نظم التشغيل المستخدمة مع الحواسيب الشخصية
يمكن تقسيم أنواع نظم التشغيل المستخدمة مع الحواسيب الشخصية إلى عدة أنواع منها :

1. نظام التشغيل (CP/M-80) Control Program

Microprocessors:

يعتبر هذا النظام نظام التشغيل القياسي للحواسيب الشخصية التي تستخدم المعالجات الدقيقة (Z80) و (8080) (المقصود بالمعالجات الدقيقة هي الميكروبروسيسورز التي تمثل عقل الحاسوب ، وهذه الأنواع من المعالجات تستخدم وحدات بيانات طولها (8 bit) .

طور هذا النوع ليصبح صالحاً للاستخدام مع بعض أجهزة أبل Apple وأجهزة راديو شاك Radio Shake .

2. نظام التشغيل (Unix)

بدأ استخدام هذا النظام عام 1969 على الأجهزة الكبيرة و المتوسطة ثم تم تعديله ليعمل على الحواسيب الشخصية ، و يعتبر هذا النظام من النظم ذات

الجودة العالية في الأداء و أقرها إلى نظام التشغيل المثالي الذي سبق شرح خصائصه بالإضافة إلى إمكانية استخدامه مع الحواسيب التي تستخدم وحدات بيانات (16 bit) و (32 bit) .

يعتمد نظام unix على ثلاث مستويات رئيسية وهي :

- البرنامج القائد Kernel وهو برنامج ينظم المهام ومخازن البيانات .
- برنامج الغلاف Shell وهو برنامج يترجم أوامر المستخدم حتى يتمكن الحاسوب من تنفيذها .
- البرامج التطبيقية والمساعدة : Tolls and Applications وهي مجموعة البرامج المستخدمة مع نظام التشغيل لتضيف له إمكانيات كثيرة .

3. نظام التشغيل (MS-DOS)

يعتبر هذا النظام أكثر نظم التشغيل انتشاراً بسبب استخدامه كنظام تشغيل لأول جهاز حاسوبي شخصي تنتجه شركة IBM ، وقد تم تعديله عدة مرات وما زال لحد الآن تظهر نماذج معدلة منه وقد بدأ بأول إصدار من البرنامج وهو (DOS 1.0) ثم استمرت الشركة المنتجة في إنتاج إصدارات متعددة حتى وصل إلى الإصدار (DOS 6,22) .

ويتميز هذا النظام بالإضافة إلى خصائص نظام التشغيل المثالي المشروحة سابقاً ، أنه سهل الاستخدام ويعمل على الأجهزة التي تستخدم المعالج الدقيق (8088) والتي تعتبر أرخص الأجهزة وتلبي متطلبات الاستخدامات الشخصية والمنزلية للحاسوب .

4. نظام التشغيل (MS-OS/2)

يعتبر هذا النظام أكثر تطوراً من نظام MS-DOS ومستخدم مع الأجهزة (PS/2) المنتجة بواسطة شركة IBM له ميزات متعددة منها استخدام النوافذ

في اختبار الأوامر المطلوبة وذلك بالإضافة إلى استخدام الطريقة المعتادة في إدخال الأوامر عن طريق سطر الأوامر بالإضافة إلى امتيازها بخصائص نظام التشغيل المثالي ما عدا إمكانية الاستخدام المتعدد .

ورغم ميزاته فإنه ليس منتشرًا أو شائع الاستخدام مثل نظام MS - DOS لكونه يعمل على الأجهزة التي تستخدم المعالج الدقيق (80286) و (80386) وهي أكثر تكلفة من المعالج (8088) .

(5 - 4) نظام التشغيل MS-DOS

كما سبق الإيضاح فإن هناك العديد من نظم التشغيل التي تختلف في خصائصها وإمكاناتها ، وقد تم اختيار نظام التشغيل DOS لدراسته في هذا المقرر لشيوعه وانتشاره بالإضافة إلى سهولة استخدامه وبساطته ، ولا يمكن لمستخدم الحاسوب الشخصي الاستغناء عنه حيث أن البرامج التطبيقية التي تعمل معه كثيرة ومتعددة .

نظام التشغيل DOS هو النظام الذي يقود عمل الحاسوب ويشكل واسطة ربط بين المستخدم الحاسوب و (MS-DOS) وهذه تمثل الحروف الأولى من الكلمة: Microsoft Disk Operating System وتعني نظام التعامل مع الأقراص ، وهذا النظام من إنتاج شركة مايكروسوفت ، وقد بدأ العمل بهذا النظام على 1981 في شركة مايكروسوفت Microsoft أو IBM لتشغيل أجهزة الحواسيب الشخصية المنتجة في شركة IBM وقد تم تعديله عدة مرات وما زال يتم تعديله حتى الآن كما ذكرنا ، وقد بدأ بأول إصدار من البرنامج باسم (MS-DOS Version 1) أي نظام التعامل مع الأقراص - الإصدار (1) ثم استمرت الشركة في إنتاج

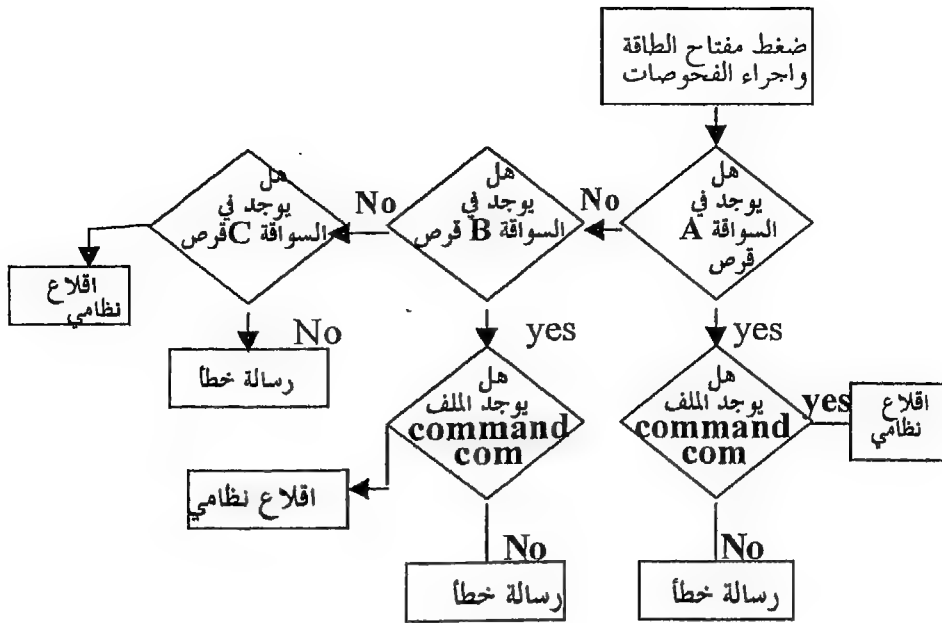
وتطوير هذا النظام حتى وصلت إلى الإصدار 6.22 (MS-DOS 6.2) وهو نظام التشغيل المستخدم في جميع أجهزة الحاسوب الشخصية.

ونشير هنا إلى أن تطوير نظام التشغيل MS-DOS ارتبط بتطوير المعالجات الدقيقة (Microprocessors) حيث أن معظم الحواسيب التي تستخدم نظام التشغيل MS-DOS تحتوي على عائلة المعالجات الدقيقة ، والتي تشتمل المعالجات (8086) (8088) (80286) (80386) (80486) .

(هذه المعالجات هي عبارة عن شرائح صغيرة (Chips) تحتوي على آلاف من الدوائر الإلكترونية المتكاملة (Integrated Circuits) التي تنتج بواسطة شركة إنتل (Intel Corporation)) .

(1-4-5) تشغيل الحاسوب

بعد الضغط على مفتاح الطاقة لتشغيل جهاز الحاسوب تقوم الروتينات الموجودة في ذاكرة ROM بإجراء فحوصات عامة على معلومات الحاسوب حيث يتم قراءة السواقة A أولاً ثم B ثانياً ثم C ثالثاً، وبعد إنهاء هذه الفحوصات يتم الإقلاع من إحدى السواقات A أو B أو C ، وفي كل الحالات يجب أن يتواجد على أحد السواقات ملف الإقلاع (Command.com) والشكل (2) يمثل إقلاع الحاسوب .



الشكل (2)

ملاحظة (1) :

في حالة وجود ملف الإقلاع على السواقة C فإنه يحذر من وضع الأقراص في السواقات الأخرى إلا بعد ظهور الرسالة التالية : Starting MS-DOS .

ملاحظة (2) :

كلمة ملف هي عبارة عن مجموعة المعلومات المرتبطة مع بعضها والتي يمكن تخزينها في الحاسوب (على أحد الأقراص) ، وأحد أهم ملفات النظام هو ملف الأوامر الذي يسمى عادة بـ Command.com و يضم الأوامر (الروتينات) الأساسية لنظام التشغيل .

ملاحظة (3) :

إذا تم الإقلاع من السواقة C عندئذ يظهر في الزاوية اليسرى العليا من

الشاشة :

C : \>

والذي يدعى محث النظام System Prompt أو محث الأوامر Command

Prompt أو مؤشر الدوس .

وإذا تم الإقلاع من السواقة A (و الذي يكون فيها قرص مرن) عندئذ

سيظهر المحث بالشكل التالي :

A: \>

و أمام محث النظام يظهر خط صغير ينبض بشكل مستمر (-) يدعى المشيرة Cursor وهي تشير مكان البدء في الكتابة على الشاشة .

(5 - 4 - 2) أوامر نظام التشغيل MS-DOS

إن نظام التشغيل MS-DOS ما هو إلا عبارة عن مجموعة من الأوامر والتعليمات التي تمكن المستخدم من التعامل مع الحاسوب مثل إدارة الملفات من نسخ وحذف وكتابة وتسمية ملف ، أو التعامل مع الأقراص وتهيئتها أو التعامل مع المكونات الأخرى للحاسوب .

تقسم أوامر نظام التشغيل MS-DOS إلى مجموعتين من الأوامر :

- الأوامر الداخلية Internal Command : وهي مجموعة من التعليمات الكثيرة الاستعمال والموجودة في ملف واحد هو Command . Com ، وهذا الملف ينتقل الى الذاكرة عند تشغيل الحاسوب ، لذلك تنفذ أي من هذه الأوامر من الذاكرة الرئيسية بدون الرجوع إلى القرص ، من هذه الأوامر ... , Copy , Date

- الأوامر الخارجية External Command : هذه الأوامر توجد على هيئة ملفات قابلة للتنفيذ ، ولتنفيذ أي أمر لا بد من طلب ملفه من القرص المخزن عليه نظام التشغيل لوضعه في الذاكرة المؤقتة RAM ومن ثم تنفيذه من هذه الأوامر, Disktopy , Format

وبشكل عام يمكن تقسم أوامر نظام التشغيل إلى أربعة مجموعات هي :

- أوامر الخدمات .

- أوامر التعامل مع الأقراص .

- أوامر التعامل مع الملفات
 - أوامر التعامل مع الفهارس (الأدلة) .
- وسنقوم فيما يلي بشرح أهم الأوامر المتعلقة بكل مجموعة، مما يلي حاجة الطالب في هذا المقرر للبدء في العمل على الحاسوب .

(1-2-4-5) أوامر الخدمات :

• الأمر VER

وظيفته : للتعرف على رقم الإصدار للنظام المعمول به على الحاسوب
صيغته :

لـ C:\> VER

الشرح : إن المقصود بالإشارة لـ هو ضغط مفتاح Enter الإدخال ،
ونتيجة تنفيذ هذا الأمر تظهر رسالة تبين رقم الإصدار للنظام المعمول به كما يلي:
MS-DOS Version 6.2

• الأمر Date

وظيفته : عرض التاريخ الحالي وإمكانية تصحيحه إذا لزم الأمر .
صيغته :

لـ C:\> DATE

الشرح : تظهر رسالة تعرض التاريخ الحالي مع إمكانية تصحيحه إذا كان
خطأً كما هو موضح في الرسالة (mm-dd-yy) أي ندخل الشهر- اليوم -
السنة ثم نضغط مفتاح الإدخال Enter ، وإذا لم يكن خطأً فإننا نضغط مفتاح
الإدخال للرجوع إلى محث النظام .

• الأمر Time

وظيفته : عرض الوقت الحالي وإمكانية تصحيحه إذا لزم الأمر .
صيغته :

لـ C:\> TIME

الشرح : تظهر رسالة تعرض الوقت الحالي فإذا كان صحيحاً نضغط مفتاح الإدخال للرجوع إلى محث النظام وإذا كان خطأ فإنه يمكن تصحيحه كما هو موضح في الرسالة : (a/p) (hours:minutes) أي الساعة ثم الدقائق ثم a للإشارة إلى التوقيت صباحي أو p للإشارة إلى التوقيت المسائي ثم نضغط مفتاح الإدخال Enter للرجوع الى محث النظام .

• الأمر CLS :

وظيفته : تنظيف الشاشة .

صيغته :

← C:\> CLS

الشرح : يؤدي إلى أنتقال محث النظام والمشيخة إلى الزاوية اليسرى العليا من الشاشة وتصبح الشاشة نظيفة من أية كتابات .

• الأمر Doskey :

وظيفته : جعل التعامل مع مؤشر DOS (محث النظام) أسهل من ذي قبل وله فوائد كثيرة :

1. يتيح استخدام مفاتيح الأسهم للانتقال الى الحرف المخطوء ثم تصحيحه فقط دون الحاجة الى مسح الحروف اللاحقة للحرف المخطوء .
 2. يتيح استرجاع الأوامر التي سبق أن أدخلت بعد تشغيل الجهاز عند مؤشر الدوس ، لذلك لا حاجة لإعادة كتابة الأوامر التي سبق إدخالها عند الحاجة إليها مرة أخرى أثناء العمل ، ولكن نستخدم مفاتيح الأسهم العلوي-السفلي لاستعراض الأوامر المدخلة إلي أن نصل إلى الأمر المطلوب إعادة كتابته :
- صيغته :

← C:\> DOSKEY

الشرح : لا شيء يحدث عند إستخدام الأمر السابق إلا أنه بواسطته يتم تعريف المفاتيح اللازمة للتحريك والاسترجاع التالية :

مفاتيح التحرير :

اسم المفتاح	وظيفته
→	الانتقال حرف واحد في كل مرة إلى اليمين
←	الانتقال حرف واحد في كل مرة إلى اليسار
Ctrl + →	الانتقال كلمة واحد في كل مرة إلى اليمين
Ctrl + ←	الانتقال كلمة واحد في كل مرة إلى اليسار
Home	الانتقال إلى بداية الأمر
End	الانتقال إلى نهاية الأمر
Esc	مسح الأمر (حذف السطر الذي كتب منذ لحظات)
Del	مسح الحرف الذي تقع عليه المشيرة
Backspace	مسح الحرف الذي يقع على يسار المشيرة
Ins	حشر الأحرف ضمن الكلمات .

مفاتيح الاسترجاع :

اسم المفتاح	وظيفته
PgUp	استرجاع أوامر الدوس الذي سبق إدخالها قبل الأمر الظاهر عند مؤشر الدوس.
PgDn	استرجاع أوامر الدوس الذي سبق إدخاله بعد الأمر الظاهر عند مؤشر الدوس
F7	استرجاع أول أمر دوس تم إدخاله
F9	استرجاع آخر أمر دوس تم إدخاله
F7	استرجاع جميع الأوامر التي سبق إدخالها مرقمة بشكل متسلسل حسب إدخالها
F9	عند الضغط عليه تظهر الرسالة : Line Number نكتب بعدها رقم أحد الأوامر التي ظهرت نتيجة استخدام المفتاح F7 ثم نضغط مفتاح الإدخال ليصار الى تنفيذ الأمر مباشرة

بعض المفاتيح الضرورية :

اسم المفتاح	وظيفته
Shift + أحد الأحرف	استخدام الأحرف والرموز المكتوبة أعلى كل مفتاح .
Ctrl + c	إنهاء عمل أي أمر
Ctrl + Z أو (Pause)	إيقاف تنفيذ أي أمر وللمتابعة نضغط أي مفتاح من لوحة المفاتيح

(5-4-2-2) أوامر التعامل مع الأقراص :

• الأمر Format :

وظيفته :

1. تشكيل (تهيئة) القرص المرن أو الصلب لكي يكون جاهزاً لحزن المعلومات.
 2. فحص القرص بحثاً عن المسارات التالفة و تحديدها لتلافي الحزن عليها .
- صيغته :

لـ : d > C:\> FORMAT

حيث d تشير إلى اسم السواقة التي تحوي القرص المراد تهيئته ، أي يمكن أن يكون a أو b أو c .

الشرح : يقوم الدوس بعملية التهيئة المكونة من ثلاثة خطوات :

1. تقسيم القرص الى مسارات (Tracks) وقطاعات (Sectors)

المسار : هو عبارة عن دائرة على القرص مركزها هو مركز القرص .

القطاع : هو المنطقة المشكلة بين مسارين ونصفي قطرين .

2. إنشاء جدول توزيع الملفات (File Allocation Table) :

جدول توزيع الملفات : هو جدول الذي سيحوي أسماء الملفات المخزنة على

القرص وعناوين تخزينها على القطاعات .

3. إنشاء دليل الجذر للقرص (Root Directory)
وعند الانتهاء من عملية التهيئة تظهر رسالة يطلب فيها إدخال اسم القرص
إذا أردنا أو نضغط مفتاح الإدخال Enter
ثم يتم عرض المعلومات المتعلقة بالقرص المهيأ وسؤال إذا كان لدينا قرص آخر
لتهيئته إما نكتب Y ثم Enter أو N ثم Enter .

ملاحظات :

1. اسم القرص يجب أن يكون مؤلف من 11 حرفاً كحد أقصى ويمكن أن يتضمن الفراغ ، غير أنه ليس بالإمكان استخدام الرموز التالية في اسم القرص:
 $\backslash ? * / : , ; = + \& ^ () [] < " >$
2. إذا قمنا بتهيئة قرص يتضمن بيانات مخزنة مسبقاً، لذلك يجب الانتباه الى خلوس القرص أو أننا لسنا بحاجة للبيانات الموجودة عليه .
3. إن وجود القطاعات التالفة في القرص لا تجعله غير صالحاً بل أن الأمر Format سيقوم بتحديددها حتى لا يقوم الدوس بخزن المعلومات عليها .

• الأمر Unformat :

وظيفته : إلغاء تهيئة الأقراص وإعادة القرص وملفاته إلى الحالة التي كانت عليها قبل عملية التهيئة .
صيغته :

د : d : \> UNFORMAT C:

الشرح : إعادة القرص إلى وضعه قبل التهيئة .

ملاحظات :

غالباً ما يستخدم هذا الأمر عندما نريد استرجاع الملفات والفهارس التي تم مسحها عند تهيئة القرص بدون قصد ، وهذا يتم فقط إذا لم نقوم بتخزين بيانات جديدة على القرص بعد. ومع ذلك يجب أن نعلم أن استخدام الأمر Unformat قد يؤدي إلى فقدان تلك البيانات أو بعضها لذلك يجب اخذ الحيلة عند استخدام

الأمر Format ، وأن نفترض دوماً أن استخدامه سيؤدي بصورة دائمة إلى فقدان البيانات الموجودة على القرص .

• الأمر VOL :

وظيفته : إظهار أسم القرص .

صيغته :

لـ : d > VOL C:\

الشرح : يعرض رسالة يوضح فيها اسم القرص إذا كنا قد سميناه مسبقاً إما عند التهيئة أو باستخدام الأمر التالي .

• الأمر Label :

وظيفته : - إعادة تسمية القرص باسم جديد دون الحاجة إلى إعادة تشكيله .

- تسمية القرص إذا لم يكن له اسم .

صيغته :

لـ : d > LABEL C:\

الشرح : يعرض الدوس رسالة يبين فيها اسم القرص (إذا كان له اسم)

ويطالب إدخال اسم جديد (إذا أردنا) ثم نضغط مفتاح Enter .

ملاحظة : الانتقال إلى القرص الموجود على إحدى السواقات نستخدم الصيغة

التالية:

الصيغة :

لـ : d > dd

حيث dd السواقة الحالية ، d السواقة التي ستنتقل إليها .

مثال : أنتقل إلى السواقة A بعد وضع قرص مرّن فيها مع العلم أن السواقة الحالية

هي C :

لـ : a > C:\

والنتيجة يصبح محت النظام بالشكل >A: والذي يمكن استخدام الأوامر عنده أيضاً .

(3-2-4-5) أوامر التعامل مع الملفات :

عندما نقوم بخزن المعلومات على الأقراص (المرنة أو الصلبة) فإن الحاسوب يقوم بحفظ المعلومات على هيئة ملف منفصل (تشبه هذه العملية تماماً عملية وضع كافة أوراق أحد التقارير في ملف واحد) وقبل أن يحفظ المعلومات يطلب منا تحديد اسم لذلك الملف كي يستطيع استرجاعه عندما نطلب منه .

يستخدم الحاسوب عادة نوعان من الملفات :

1. ملفات نصية Text File : تحتوي على محارف ورموز يمكن قراءتها .
2. ملفات أوامر Command File : وتحتوي على مجموعة من الأوامر التي يحتاجها النظام DOS لتنفيذ أوامره ومحتواها غير قابلة للقراءة لأنها مكتوبة بلغة الآلة .

يتكون اسم الملف من جزئين رئيسيين يفصل بينها نقطة (.) dot .

الجزء الأول : اسم الملف file name ويتألف من ثمانية محارف على الأكثر .

الجزء الثاني : الامتداد Extension ويتألف من ثلاثة محارف على الأكثر .

ويمكن استخدام كافة الحروف (الصغيرة و الكبيرة) والأرقام والرموز باستثناء الرموز التالية :

? ; ! < * + : [\ / . الفراغ

وللامتداد فائدة كبيرة للحاسوب إذ يساعد على تمييز نوع الملف ، إذ أن الحاسوب يتعامل عادة مع امتدادات نموذجية نذكر منها :

<u>الامتداد</u>	<u>يعبر عن ملف</u>
LTR	رسالة
TXT	نص
DWG	برنامج رسم
DOC	وثيقة Document
COM	برنامج أوامر
EXE	برنامج قابل للتنفيذ
SYS	برنامج تابع للنظام
HLP	برامج مساعد
BAS	برنامج مكتوب بلغة البيسك
PAS	برنامج مكتوب بلغة باسكال
FOR	برنامج مكتوب بلغة الفورتران
BAT	برنامج تنفيذي تدفقي

• الأمر DIR :

وظيفته : استعراض الملفات الموجودة على القرص (المرن أو الصلب) .
صيغته :

(1) DIR >C:\

(2) DIR d >C:\

الشرح : إن الصيغة الأولى تؤدي إلى استعراض جميع الملفات الموجودة على القرص الصلب و الصيغة الثانية تؤدي إلى استعراض جميع الملفات الموجودة على القرص المرن الموجود في السواعة d (حيث d إما A أو B) والعودة إلى محث النظام >C:\ .
إن الاستعراض يتضمن أسماء الملفات والامتدادات - سعة كل ملف - تاريخ ووقت إنشاء الملف أو آخر تعديل على الملف في نهاية الاستعراض يعرض عدد الملفات وسعتها الإجمالية والسعة المتعلقة على القرص .

ملاحظات :

1. غالباً ما يكون عدد الملفات كبيراً ولا يتسع على الشاشة وفي هذه الحالة الأمر DIR سيعرض آخر مجموعة ملفات (أي الصفحة الأخيرة ، حيث اعتبرنا أن سعة الشاشة تمثل صفحة ، وبالتالي لاستعراض الملفات صفحة صفحة نستخدم الصيغة .

C:\> DIR /P ↵

C:\> DIR /P d: ↵

2. لاستعراض فقط جميع أسماء الملفات بدون السعة والتاريخ والوقت وعلى صفحة واحدة نستخدم الصيغة :

C:\> DIR /W ↵

C:\> DIR /W d: ↵

3. لاستعراض جميع الملفات ذات صفة مشتركة نستخدم :

إما (*) كبديل عن مجموعة أحرف قد تصل إلى ثمانية أحرف .

أو (?) كبديل عن حرف واحد .

أمثله : - استعراض جميع الملفات على السواقة C ذات الامتداد TXT :

C:\> DIR *.TXT ↵

- استعراض جميع الملفات على السواقة A ذات الاسم Command :

C:\> DIR A: Command.* ↵

- استعراض جميع الملفات على السواقة C التي يكون الحرف الأول من

الاسم مجهول وبقية الاسم TEL .

C:\> DIR ?TLE.* ↵

• الأمر Copy :

وظيفة : (1) إنشاء ملف نصي

صيغته :

C:\> copy con اسم الملف

الشرح : الأمر السابق مفاده الطلب من نظام الدوس نسخ ما يجري إدخاله من لوحة المفاتيح ووضعها تحت اسم الملف المعطى .
وتنفيذ الأمر السابق يؤدي إلى ظهور المشيرة على السطر التالي ، نبدأ مباشرة بكتابة أي نص باللغة الإنجليزية ، وإنهاء الكتابة والتخزين نستخدم المفتاحين Ctrl + Z ، ثم مفتاح Enter.

ملاحظة :

1. يعتبر هذا الأمر أبسط محرر نصوص يعمل تحت نظام التشغيل MS-DOS ولكن مساوئه أنه لا يمكننا من العودة الى السطر السابق للكتابة أو التصحيح بعد الضغط على مفتاح الإدخال ENTER بالإضافة الى أننا لا يمكننا إضافة أو حذف أي حرف بعد تخزينه .

2. إذا أردنا إنشاء ملف نصي على قرص مرن موجود في إحدى السواقات نضيف فقط قبل اسم الملف اسم السواقة ونقطتين رأسيين كما يلي :

لـ الإمتداد . اسم الملف a: Copy Con >C:

وظيفة : (2) نسخ ملف من سواقة إلى سواقة أخرى أو إلى نفس السواقة مع

تغيير اسم الملف صيغته العامة :

صيغته :

لـ الإمتداد . اسم الملف الجديد d2: الامتداد . اسم الملف d1: Copy >C:

الشرح : يقوم النظام DOS بنسخ محتوى الملف الموجود على القرص

الموجود في السواقة d1(C,B,A) إلى القرص الموجود في السواقة d2(C,B,A)

تحت اسم جديد .

ويمكن استخدام اسم الملف الأصلي إذ كنا ننسخه إلى قرص موجود في سواقة أخرى .

أمثلة :

- 1 - C:\>COPY C:MM.TXT a: لـ
- 2 - C:\>COPY a:MM.XTX a:MAX.TXT لـ
- 3 - C:\>COPY *.TXT a:*.BAK لـ
- 4 - C:\>COPY a:Come.doc C: لـ

وظيفة : (3) جميع مغلقين أو أكثر في ملف واحد

صيغته :

لـ الامتداد. اسم الملف الجديد الامتداد. اسم الملف (2) + الامتداد . اسم الملف (1)

C:\>COPY

الشرح : يقوم الـ DOS بجمع الملفين (1) و (2) في ملف تحت اسم الملف الجديد.

ملاحظات :

(1) إذا كانت جميع الملفات في أماكن مختلفة ندخل قبل اسم الملف السواعة التي

تحتوي القرص الذي يحوي الملف المطلوب مع نقطتين رأسيتين (:). كما يلي :

C:\>Copy a:Game.TXT + C:Prog.TXT a:MM.doc لـ

(2) إذا لم نكتب اسم الملف الجديد فإن الملف الناتج عن الجمع يأخذ اسم الملف الأول وامتداده .

(3) يمكن جمع أكثر من ملفين بنفس الأسلوب .

• الأمر Move :

وظيفته : نقل ملف .

صيغته :

الامتداد . اسم الملف :d2 الامتداد . اسم الملف :d1 Move C:\>

الشرح : يقوم الدوس بحذف الملف الأصلي من القرص الموجود في السواعة

d1 وتخزينه على القرص الموجود في السواعة d2 إما باسم الملف الأصلي حسب

ما ندخل .

مثال :

C:\>Move a:MM.TXT C: \

C:\>Move a:MM.TXT C: NCX.TXT \

• الأمر DEL :

وظيفته : حذف الملف .

صيغته : \> DEL d اسم الملف :

الشرح : يقوم الدوس بحذف الملف الموجود على القرص الموجود في السواعة (C,B,A)

ملاحظات :

1- إذا لم نضع أمام اسم الملف اسم السواعة يقوم الدوس بالتفتيش عن الملف المطلوب في السواعة C ويحذفه.

2- يمكن إجراء حذف مجموعة ملفات باستخدام * أو ؟

C:\>DEL a:*.TXT \

3- عملية الحذف تؤدي الى حذف الملف نهائياً ويمكن استرجاعه أحياناً إذا لم نقوم بأي عملية على القرص الذي يحوي الملف المحذوف وذلك باستخدام الأمر التالي :

• الأمر Undelete

وظيفته : إلغاء حذف الملف شريطة أن لا نكون قد قمنا بتخزين أو نسخ على القرص الذي كأن يحوي الملف المحذوف .

صيغته :

\>UNDELETE D اسم الملف :

الشرح : سيعرض الدوس رسالة يظهر فيها اسم الملف ما عدا الحرف الأول

ثم يعرض رسالة يسأل فيها إذا كنا نريد فعلاً إلغاء عملية الحذف أم لا ، فإذا كنا

نريد فعلاً إلغاء عملية الحذف واسترجاع الملف المحذوف نضغط على الحرف Y (أي Yes) ثم نكتب الحرف الأول من اسم الملف ثم نضغط على مفتاح ENTER فيعرض الدوس سؤال آخر إذا كنا نريد إلغاء حذف ملف آخر وعندئذ نضغط على Y ونكتب الحرف الأول من اسمه أو نضغط N .

ملاحظة هامة : عندما نقوم بحذف الملفات فإن الدوس في الواقع لا يقوم بحذفه نهائياً بل يقوم بحذف الحرف الأول من اسمه وبذلك يهيئ المساحة التي كان يشغلها الملف المحذوف على القرص لحزن أي ملف لاحق نقوم بحزنه ، وما لم نقوم بحزن أي ملف على القرص بعد إجراء عملية حذف الملفات فإنه لدينا فرصة لإلغاء عملية الحذف .

• الأمر Type :

وظيفته : استعراض محتوى ملف معين .

صيغته :

الامتداد اسم الملف : d Type >C:

الشرح : استعرض محتوى الملف الموجود على القرص الموجود على السقاقة (A أو B أو C) .

وإذا لم نكتب اسم السقاقة فإنه يقوم الدوس بالتفتيش عن الملف المكتوب في السقاقة C لاستعراضه .

ملاحظات :

1. الملفات ذات اللاحقة EXE أو COM ستكون مكتوبة بلغة الآلة وبالتالي عند استعراض تلك الملفات تظهر شفرة غير مقروءة .

2. إذا كان سعة الملف أكبر من سعة الشاشة عندئذ يمكن استخدام الأمر التالي لاستعراض جزءاً جزءاً من الملف .

الامتداد اسم الملف : d TYPE >\C: | MORE

حيث | أحد مفاتيح لوحة المفاتيح .

• الأمر Print :

وظيفته : طباعة محتوى ملف على الطابعة .

صيغته :

لـ الامتداد . اسم الملف d: >PRINT C:\

الشرح : يتم نقل محتوى الملف المطلوب إلى الطابعة ليصار إلى طباعته على الورق.

• الأمر Ren :

وظيفته : إعادة تسمية الملف .

صيغته :

لـ الامتداد.الاسم الجديد للملف الامتداد.اسم الملف d: >REN C:\

الشرح : يقوم الدوس بتغيير الاسم القديم للملف الموجود على القرص الموجود على السواقة d (حيث d يمكن أن يكون A,B,C) الى الاسم الجديد.

وإذا كان الملف على السواقة C فلا حاجة لكتابة (C) قبل اسم الملف .

ملاحظة : يمكن إعادة تسمية مجموعة من الملفات وذلك باستخدام * أو ؟

لـ مثال : C:\>REN *.TXT *.DOC

• الأمر Attrib :

وظيفته : إظهار الخواص المميزة للملف مع إمكانية التعديل في هذه الخواص

(إضافة أو حذف) .

صيغته :

1. التعرف على شكل الملف

الامتداد . اسم الملف لـ d: >Attrib C:\

2. تغيير الصفة :

لـ الامتداد . اسم الملف الصفة (- أو +) C:\> Attrib

الشرح : أي ملف يمتلك صفات معينة ترشد الحاسوب ونظام التشغيل إلى كيفية التعامل مع هذا الملف وهذه الصفات هي :

رمز الصفة	اسم الصفة	وظيفتها عندما تكون مصاحبة للملف
R	Read Only قراءة فقط	لا يمكن تعديل الملف أو الكتابة أو حذفه ولكن فقط القراءة
H	Hidden الإخفاء	إخفاء الملف من قائمة الملفات عند استخدام أمر Dir .
S	System نظامي	الملف مخصص لاستخدامات النظام .
A	Archive الأرشفة	تشير إلى أن الملف جديد أو أنه قديم ويحتاج في كلا الحالتين إلى نسخة احتياطية .

إن استخدام صيغة التعرف على الملف تؤدي إلى ظهور الرسالة ذات الشكل التالي:

الامتداد . اسم الملف \ اسم السواعة

أما استخدام الصيغة الثانية فأن إشارة + تشير إلى إضافة الصفة ، وإشارة (-) تشير إلى حذف الصفة.

ملاحظات :

(1) يمكن استخدام الصيغتين السابقتين من أجل مجموعة من الملفات باستخدام * أو؟

(2) يمكن إجراء عملية الحذف والإضافة لأكثر من صفة في آن واحد .

(3-4-5) أوامر التعامل مع الفهارس :

لنستعرض المثال التالي :

بفرض أنه لدينا في قسم الأرشفة لشركة ما الملفات التالية:

- 1-ملفات خاصة بالموظفين وتضم :ملفات خاصة بالحضور والغياب .
ملفات خاصة بالرواتب والأجور والتعويضات .
ملفات خاصة بالعقوبات .
- 2-ملفات خاصة بالإدارة وتضم :ملفات خاصة بالتعليمات .
ملفات خاصة بالشؤون الإدارية.
- 3-ملفات خاصة بالمعدات وتضم :ملفات خاصة بالأجهزة .
ملفات خاصة بقطع التبديل .

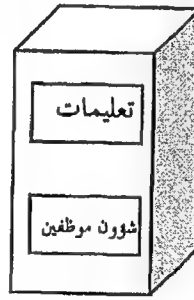
عندئذ أمام موظف الأرشيف أحد الخيارين :

الخيار الأول : وضع الملفات فوق بعضها البعض دون ترتيب ، وفي هذه الحالة عندما يطلب منه إحضار ملف معين ، فإنه يضطر إلى البحث في كل الملفات كي يجد طلبه له، وهذا ربما يستهلك وقتاً وجهداً كبيرين .

الخيار الثاني : ترتيب الملفات وتنظيمها ضمن خزائن وأدراج كما في الشكل (3).



ملفات خاصة بالموظفين



ملفات خاصة بالإدارة



ملفات خاصة بالمعدات

الشكل (3)

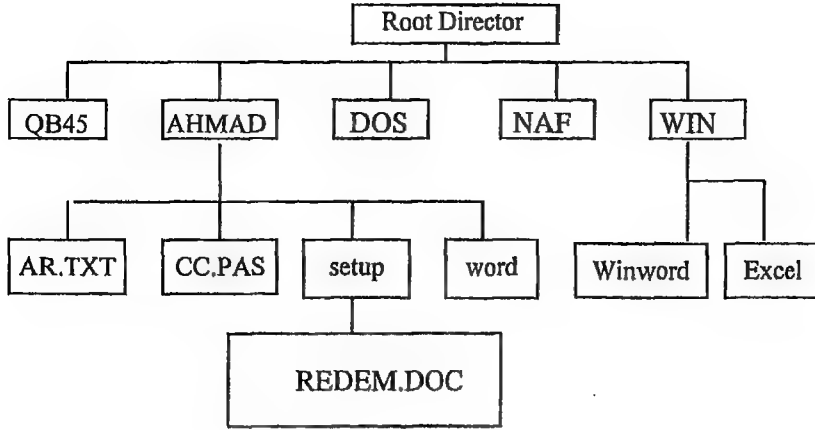
وفي هذه الحالة عندما يطلب منه إحضار ملف معين تصبح العملية سهلة حيث يتوجه مباشرة إلى الخزانة التي تحوي هذا النوع من الملفات ون ثم إلى الدرج المناسب وبذلك يختصر وقت وجهد كبيرين .

إن الخيار الثاني في تنظيم غرفة أرشيف الشركة يشبه تماما عملية تخزين الملفات وتنظيمها بواسطة نظام التشغيل سواء كان التخزين على القرص المرن أو القرص الصلب ولكن باستخدام الفهارس (أو الأدلة) (Directories) والفهارس الفرعية Subdirectories التي يقوم نظام التشغيل MS-DOS بإنشائها باستخدام مجموعة من الأوامر .

يمثل القرص (الصلب أو المرن) الفهرس الجذر (الرئيسي) Root Directory الذي يحتوي على ملفات وفهارس فرعية ، والفهارس الفرعية هي عبارة عن منطقة تخزينية من القرص لها اسم معين ويمكن أن تحوي على ملفات وفهارس فرعية أيضا ، وعندما نقوم بتهيئة أي قرص جديد باستخدام الأمر Format فإن نظام الدوس يقوم بإنشاء فهرس الجذر ويعتبر هذا الفهرس هو فهرس افتراضي لحين الانتقال إلى فهرس آخر ، ونظام الدوس يخصص رمز الشوطة المائلة \ للفهرس الجذر ولا يسمح نظام الدوس بحذف الفهرس الجذر ، وأما الفهرس الفرعي فهو أي فهرس باستثناء الفهرس الجذر.

اسم الفهرس يتألف من ثمانية محارف كما في تسمية الملفات باستثناء الامتداد وعند استعراض الملفات بالأمر DIR فإنه يتم عرض بالإضافة الى الملفات أسماء الفهارس الفرعية الموجودة ضمن الفهرس الجذر وأمام الأسم مكتوب <DIR> .

والشكل (4) يبين التركيب الهيكلي للفهارس والملفات في قرص على شكل شجرة مقلوبة (الشجرة افتراضية) :



الشكل (4)

ل للوصول الى الملف REDEM.DOC الموجود في الفهرس الفرعي SETUP المتفرع عن الفهرس الفرعي AHMAD المتفرع عن الجذر في القرص الموجود في السواعة A نكتب عند مؤشر الدوس ما يلي :

C:\>{أحد أوامر الدوس} A:\AHMAD\SETUP\REDEM.DOC

↑ السواعة الحالية ↑ اسم السواعة الموجودة فيها القرص ↑ اسم السواعة ↑ فهرس فرعي ↑ فهرس فرعي ↑ ناصلة ما بين الفهارس ↑ ناصلة ما بين الفهارس ↑ الملف المطلوب

إن سلسلة الفهارس والفهارس الفرعية التي يتوجب على نظام الدوس سلوكها للوصول الى الملف المطلوب يدعى المسار (PATH).

وعند الضغط على مفتاح الإدخال ENTER للشكل السابق فسوف يقوم الدوس بالبحث في القرص الموجود في السواعة A وذلك ابتداء من الفهرس الجذر \ حيث يبحث عن الفهرس الفرعي AHMAD ومن ثم يبحث ضمن الفهرس الفرعي AHMAD عن الفهرس الفرعي SETUP وبعدئذ يبحث ضمن الفهرس الفرعي SETUP عن الملف المطلوب REDEM.DOC ، بعد ذلك يطبق على الملف المطلوب أمر الدوس المعطى عند محث النظام مباشرة (أمر الدوس أي أمر من أوامر الدوس التي مرت معنا سابقاً أو التي ستمر لاحقاً) .

ملاحظة (1) : يوجد طريقة ثانية للوصول الى الملف وذلك بالانتقال الى الفهرس الفرعي الذي يحوي الملف المطلوب بالأمر (CD) الذي سنشرحه لاحقا ، ثم نطبق الأمر على الملف ، بمعنى آخر يمكن تنفيذ جميع الأوامر المتعلقة بالنظام دوس ضمن الفهرس الفرعي وكأن الفهرس الفرعي قرص مستقل بحد ذاته .

• الأمر MD :

وظيفته : إنشاء فهرس MAKE DIRECTORY

صيغته : (1) إنشاء فهرس فرعي على القرص C :

C:\>MD اسم فهرس فرعي

(2) إنشاء فهرس فرعي على القرص الموجود في إحدى السواقات B,A وسنرمز لها بـ d .

C:\>MD d: اسم فهرس فرعي

(3) إنشاء فهرس فرعي موجود ضمن القرص الموجود في أحد السواقات d .

C:\>MD d: \المسار اسم فهرس فرعي

ويمكن أيضا الانتقال الى السواقة المطلوبة والفهرس الفرعي المطلوب ثم إنشاء فهرس فرعي كما سننشى ذلك في الأمثلة التابعة للأمر CD.
أمثلة :

1. أنشئ ثلاثة فهرس فرعية في القرص المرن الموجود في السواقة A وهي :

(GAMES,PROGRAM, AHMAD)

C:\>MD A:AHMAD

C:\>MD A:PROGRAM

C:\>MD A:GAMES

2. أنشئ فهرسين ضمن الفهرس الفرعي AHMAD باسم AHMAD1

AHMAD2

C:\>MD A:AHMAD\AHMAD1

C:\>MD A:AHMAD\AHMAD1

3. أنشئ ثلاثة فهارس فرعية ضمن الفهرس الفرعي AHMAD1 باسم
SS1,SS2,SS3

C:\>MD A:AHMAD\AHMAD1\SS1

C:\>MD A:AHMAD\AHMAD1\SS2

C:\>MD A:AHMAD\AHMAD1\SS3

• الأمر CD :

وظيفة : الانتقال الى الفهرس فرعي آخر Ching Directory

صيغته :

1. للانتقال الى الفهرس الفرعي الأبن (أي فهرس فرعي داخل فهرس معين)

C:\> CD d: اسم الفهرس الفرعي

C:\> CD d: اسم الفهرس الفرعي

2. للانتقال الى الفهرس الاب (أي الفهرس الفرعي الذي يضم فهرس معين)

C:\> CD ..

3. للانتقال الى الفهرس الجذر مباشرة نكتب :

C:\> CD \

أمثله : 1. أنشئ فهرسين فرعين ضمن الفهرس Program الموجود على السواعة

A باسم Program1 و Program2 وذلك بعد الانتقال إلى الفهرس

Program مع العلم أن السواعة الحالية C .

الخطوة (1) الانتقال إلى السواعة A وذلك :

C:\> A:

ينتج محث النظام بالشكل A:\>

الخطوة (2) الانتقال إلى الفهرس الفرعي Program وذلك :

```
A:\> CD Program ↵
```

ينتج محث النظام بالشكل ↵ A:\> Program>

الخطوة (3) إنشاء الفهرسين الفرعين المطلوبين بالتالي أي :

```
A:\> Program> MD Program 1 ↵
```

```
A:\> Program> MD Program 2 ↵
```

3. أنشئ ثلاثة فهارس فرعية ضمن الفهرس الفرعي Program1 باسم
ss3,ss2,ss1 (ترك للطالب) .

• الأمر Tree :

وظيفته : عرض شجرة الفهارس للسواقة الحالية .

صيغته :

```
dd:\> Tree ↵
```

حيث dd السواقة الحالية المراد عرض شجرتها .

```
dd:\> Tree d: ↵
```

حيث d السواقة التي تحوي القرص المراد عرض شجرته .

أمثله : 1. استعرض الفهارس الفرعية على السواقة C :

```
C:\> Tree ↵
```

3. استعرض الفهارس الفرعية على السواقة A .

```
C:\> Tree A: ↵
```

```
C:\>A: ↵ } أو  
A:\> Tree: ↵ }
```

ملاحظات :

1- لعرض شجرة الفهارس مع الملفات :

dd:\> Tree d:\F ↓

حيث dd السواقة الحالية ، d السواقة المراد عرض شجرتها وملفاتها وعندما dd=d لا حاجة لكتابة d .

2- إذا كان الشجرة لا تتسع في شاشة واحدة يمكن استخدام الأمر التالي للاستعراضها جزءاً جزءاً .

dd:\> Tree d:\F |More ↓

3- لعرض جميع الملفات و الفهارس الموجودة في الفهرس الحالي الأمر DIR كما مر معنا سابقا .

4- إن جميع الأوامر السابقة التي مرت معنا يمكن استخدامها انطلاقاً إما من السواقة الحالية وذلك بتحديد المسار أو بالانتقال الى الفهرس الفرعي المطلوب أينما كان باستخدام CD .

أمثلة :

1. أنشيء ملف نصي في الفهرس الفرعي Ahmad1 باسم MM.TXT

↓ C:\> Coy Con A:Ahamd\Ahmad1\MM.TXT

ويمكن انشاء الملف بالطريقة التالية :

أوامر الدوس المتتالية	نتيجة التنفيذ على الشاشة
1) C:\> A: ↓	A:\>
2) A:\> CD Ahmad: ↓	A:\> Ahmad>
3) A:\> Ahmad>CD Ahmad1 ↓	A:\> Ahmad> Ahmad1>
4)A:\> Ahmad>Ahmad1> copy con MM.TXT ↓	ونبدأ بكتابة نص بعد ظهور مؤشر الكتابة على السطر

2- انسخ الملف MM.TXT من الفهرس الفرعي Ahmad1 إلى الفهرس الفرعي ss1 الموجود ضمن الفهرس الفرعي Ahmad1 انطلاقاً من السوافة C.

```
C:\>Copy A:\Ahmad\Ahmad1\MM.TXT  
\\ss1\MM.TXT ← a:\Ahmad\Ahmad1
```

3- انسخ الملف MM.TXT من الفهرس الفرعي Ahmad إلى الفهرس الفرعي SS1 الموجود ضمن الفهرس الفرعي Program انطلاقاً من C وباسم جديد LS.DOC .

```
C:\>Copy A:\Ahmad\Ahmad1\MM.TXT  
a:\Ahmad\Ahmad1\SS1\LS.DOC ←
```

• الأمر RD :

وظيفته : حذف أحد الفهارس الفرعية .

صيغته :

```
dd:\> RD اسم الفهرس المراد حذفه
```

إلا أنه لا يمكن تطبيق هذا الأمر مباشرة قبل حذف جميع الملفات و الفهارس الفرعية الموجودة داخله لذلك يجب اتباع ما يلي عند حذف أحد الفهارس الفرعية .

1. الانتقال الى الفهرس الفرعي المطلوب حذفه بالأمر CD .
2. حذف محتويات هذا الفهرس الفرعي المطلوب باستخدام الأمر del *.* .
3. حذف الفهارس الفرعية إن وجدت فيه .
4. الرجوع إلى الفهرس الذي يحتوي الفهرس المطلوب (أي الفهرس الأب) بواسطة الأمر ..CD .
5. تنفيذ حذف الفهرس المطلوب بالأمر : اسم الفهرس المطلوب حذفه RD

مثال : أحذف الفهرس الفرعي Program مع العلم أن السواقة الحالية C .

أوامر الدوس المتتالية	نتيجة التنفيذ على الشاشة
1) C:\> A: _	A:\>
2) A:\> CD Program _	A:\> Program>
3) A:\> Program>CD program1 _	A:\> Program>program1>
4) A:\>Program>Program1>del *.* _	حذف جميع الملفات الموجودة تحت الفهرس Program1
5) A:\>Program>Program1>rd ss1 _	حذف الفهرس الفرعي ss1 الموجود تحت الفهرس Program1
6 A:\>Program>Program1>cd.. _	A:\> Program>
7) A:\>Program>RD Program1 _	حذف الفهرس Program1

• الأمر DELTREE :

وظيفته : حذف الفهرس الفرعي وفروعه وملفاته بخطوة واحدة وذلك باتباع ما يلي :

- 1- الانتقال إلى الفهرس الذي يحوي الفهرس المطلوب حذفه (الأب)
- 2- تكتب الأمر بالشكل التالي

اسم الفهرس المطلوب حذفه DELTREE >الفهرس الأب

مثال : المثال السابق ينفذ بالشكل التالي بعد الانتقال إلى الفهرس الفرعي

Program

A:\> Program > DELTREE Program1

عندئذ يسأل الدوس إذا كنا فعلا نريد إزالته فنضغط Y وإلا نضغط مفتاح N. هذا الأمر خطير إذا لا مجال للتراجع بينما في الأمر السابق تعدد الخطوات تجعل عملية الحذف بطيئة ويكون هناك مجال للتفكير في الحذف .

• الأمر DISKCOPY :

وظيفة : نسخ محتويات القرص بأكمله إلى قرص آخر
صيغة :

لـ : d1: d2: >DISKCOPY C:\:

حيث d1 يمثل اسم السواعة التي تحوي القرص المراد النسخ منها (القرص الأصلي)

d2 يمثل اسم السواعة التي تحوي القرص المراد النسخ إليه (القرص المستقبل)

ملاحظات :

- 1- قبل البدء بعملية نسخ الأقراص يجب التأكد من أن القرص المستقبل خالياً من أية بيانات وحجمه وسعته تطابق الحجم والسعة في القرص الأصلي .
- 2- ليس بالضرورة أن يكون القرص المستقبل مهياً بل إن الأمر نفسه يهيئه أتوماتيكياً.
- 3- يجب حماية القرص الأصلي من إعادة الكتابة عليه (بفتح البوابة الموجودة في الزاوية السفلي من القرص المرن) .
- 4- إذا كان لدينا سواعة واحدة للأقراص المرنة فإن العملية تتم بالشكل التالي:
نكتب الأمر كما يلي :

لـ : A: A: >DISKCOPY C:\:

سيبدأ النظام دوس بنسخ القرص الأصلي ويضع المعلومات المنسوخة في ذاكرة الحاسوب وحين تمتلئ الذاكرة ستظهر رسالة تطلب منك إدخال القرص المستقبل في السواعة .

- نسحب القرص الأصلي من السواعة ونضع القرص المستقبل مكانه ونضغط على مفتاح الإدخال ENTER عندئذ سيقوم الدوس بنقل البيانات الموجودة في الذاكرة إلى القرص المستقبل .

- إذا كان حجم المعلومات الموجودة في القرص الأصلي اكبر من سعة الذاكرة عندئذ ستظهر رسالة تطلب منك إدخال القرص ثم نضغط ENTER

وسيقوم الدوس مرة ثانية بنقل بقية المعلومات إلى ذاكرة الحاسوب ثم يطلب إدخال القرص المستقبل لاستكمال عملية النسخ .

- وهكذا تستمر المبادلة إذا لزم الأمر لحين إتمام عملية نسخ القرص الأصلي وظهور رسالة يسأل فيها الدوس إذا كنا نريد نسخ قرص آخر أم لا فنضغط Y في حالة الإيجاب ، ونضغط على N في حالة النفي .

5- في حالة وجود سواقتين للأقراص المرنة في الحاسوب متشاهتين بالحجم والسعة فإن عملية النسخ تتم بسهولة وأسرع دون الحاجة إلى مبادلة الأقراص.

(5-5) نظام التشغيل Windows

(5-5-1) تعريف وميزات النظام ويندوز :

يعرف نظام التشغيل ويندوز على أنه نظام مواجهة رسومية Graphical User Interfaces يقوم بدور الوسيط بين المستخدم والحاسوب ، يستخدم هذا النظام أشكالاً ورموزاً صغيرة تدعى الأيقونات Icons ، هذه الأيقونات تمثل تطبيقات معينة أو عمليات تتعلق بالتطبيقات بالإضافة إلى استخدامه لقوائم الأوامر Menus لتسهيل عمليات اختيار وتشغيل الأوامر عن طريق الفأرة Mouse

ولهذا النظام مزايا عديدة أهمها :

- 1- يتيح مشاهدة النص والرسوم على الشاشة تماماً كما ستظهر عند طباعتها.
- 2- يتيح تشغيل عدة برامج تطبيقية في آن واحد ، وكل برنامج في إطاره الخاص، على عكس نظام التشغيل دوس MS-DOS الذي يتيح تشغيل برنامج تطبيقي واحد في كل مرة .

3- جميع البرامج التطبيقية التي تعمل داخل النظام ويندوز تخضع لنفس القواعد والخطوات من حيث التشغيل والخروج من البرنامج والتحويل بين البرامج المختلفة وتكبير وتصغير ونقل الأطر ، مما يسهل تعلم البرامج التطبيقية الجديدة ، لذلك فالمهارات التي يكتسبها المستخدم بالعمل على النوافذ

الأساسية لنظام التشغيل ويندوز تسهل عليه الانتقال إلى البرامج التطبيقية المختلفة التي تعمل تحت هذا النظام .

4- يعطي إمكانية تناقل البيانات بين البرامج التطبيقية المختلفة التي تعمل تحت هذا النظام ، بكلمة أخرى إمكانية نقل وتحويل وربط النصوص والبيانات والرسوم بين البرامج التطبيقية المختلفة.

5- يعتمد نظام التشغيل ويندوز على استخدام الرسومات والأشكال في إدخال الأوامر للحاسوب بدلاً من كتابتها من خلال سطر الأوامر كما في نظام التشغيل دوس .

وقد أنتجت شركة Microsoft العديد من الإصدارات المختلفة لبرامج النظام ويندوز وكل إصدار مطور وأفضل من السابق في كثير من المميزات والمواصفات .

من تلك الإصدارات هي أنظمة التشغيل ويندوز Windows 3.x والتي كانت آخرها الإصدار 3.11 عام 1994 (Windows 3.11)

ثم تلى ذلك نظام ويندوز 95 الذي يعتبر نقله نوعية في عالم البرمجيات ، إذ يعتبر نظام تشغيل تعددي المهام له القدرة على دعم أنواع الشبكات ويسمح بالاتصال بالإنترنت ويعطي بنية جديدة لنظام الملفات تسمح بوصول أطوال الأسماء الى 256 رمزاً ، بالإضافة إلى أنه نظام تشغيل ذي 32 بتا يتمتع بواجهة استخدام مريحة وجذابة وفعالة في آن واحد ، وسوف نتطرق في الفقرات اللاحقة إلى بعض المبادئ الأساسية للعمل على هذا النظام .

ثم ظهر ويندوز 97 (Windows 97) الذي عالج بعض المشاكل التي كان يعاني منها ويندوز 95 ، ثم ظهر ويندوز 98 (Windows 98) كتطوير لنظام ويندوز 95 ، في الحقيقة لا يوجد فرق يذكر بين ويندوز 98 ويندوز 95 ، فالشاشة الرئيسية والشاشات الفرعية في نوافذ 98 تشابه مثيلاتها في نوافذ 95 مع

بعض الإضافات الطفيفة ، لذلك فمستخدمي ويندوز 95 يمكنهم التعامل مع ويندوز 98 بكل سهولة وبدون أية مشاكل .

لقد جاء ويندوز 98 لتسهيل التعامل مع معدات الحاسوب Hardware

مثل :

1. تسهيل عمليات نقل البيانات من وإلى الذاكرة الرئيسية .
2. ضمان سرعة أكبر للجهاز .
3. توفير حيز أكبر للتخزين .
4. تسهيل عمليات التعامل مع شبكة الإنترنت والبريد الإلكتروني .
5. تصحيح أعطال الملفات بشكل أوتوماتيكي حين حدوثها .
6. دعم مشغلات الأقراص الصلبة ذات السعة العالية .
7. التعرف على معدات الحاسوب المضافة بشكل أفضل .
8. تقليل الأخطاء الناجمة عن المعدات والبرمجيات .
9. توفير قوائم مساعدة (Help) متقدمة وشاملة .

(2-5-5) المبادئ الأساسية للتعامل مع النظام ويندوز 95

Controlling Windows 95

أولاً : تشغيل ويندوز 95 : Starting Windows 95

وذلك بالضبط على زر التشغيل Power ، عندئذ يظهر شعار
Microsoft Windows الذي يبقى على الشاشة بضع ثوان ، ثم تظهر نافذة
تدعى نافذة التشغيل كما في الشكل (5) ، تحوي هذه النافذة على أيقونات البرامج
وشريط المهام وزر البدء ...

أما الأيقونات التي عادة تظهر عند تشغيل ويندوز 95 هي :

1. أيقونة جهاز الكمبيوتر My Computer : وتضم داخلها أيقونات فرعية
أخرى :

- أيقونة محرك القرص المرن A : تفيد في استعراض محتويات القرص
المرن (A) .

- أيقونة نظام التشغيل MS-DOS : للانتقال إلى شاشة الدوس .

- أيقونة لوحة التحكم Control Panel : للتحكم بسرعة الفأرة
والألوان الشاشات وشاشة التوقف ..

- أيقونة الطابعات Printer : للتحكم بالطابعة المتصلة مع الجهاز .

- أيقونة الاتصال الشبكي الهاتفي .

2. أيقونة الإنترنت The Internet : للاتصال بشبكة الإنترنت .

3. أيقونة Setup the Microsoft Network

4. أيقونة سلة المحذوفات Recycle Bin ، يتم داخلها حفظ الملفات التي يتم

حذفها ، هذه الأيقونة تعطي إمكانية استرجاع ملف محذوف سابقاً ، وعادة
تبقى الملفات المحذوفة داخلها ما لم نقوم بإفراغ السلة .

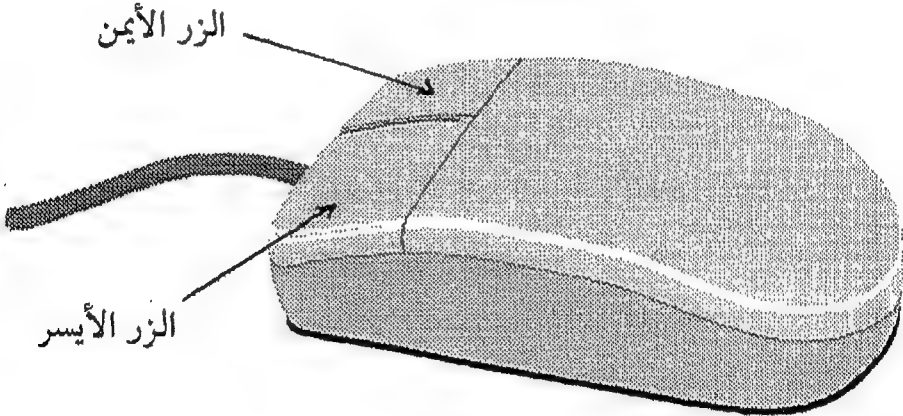
5. أيقونة حقيبة الملفات My Briefcase .

6. بالإضافة الى ايقونات برامج تطبيقية أخرى ويتعلق ذلك بالمستخدم نفسه والبرامج التي تهمه بشكل دائم .

ثانياً : استعمال الفأرة Using Mouse

تقوم الفأرة بالكثير من المهام فهي الأداة الرئيسية فيه ، إذ أنها توفر على المستخدم عناء إدخال الأوامر من لوحة المفاتيح ، وتزيد من سرعة وفعالية العمل ونستطيع بالفأرة الوصول إلى كل زوايا الشاشة والتنقل بحرية في كل مكان مما يجعل منها أداة مثالية في نظم التشغيل ذات الواجهات الرسومية مثل النظام ويندوز ..

تتألف الماوس التي تستخدم مع ويندوز 95 من زرین (الزر الأيسر والزر الأيمن) أنظر الشكل (6) .



الشكل (6) أجزاء الفأرة

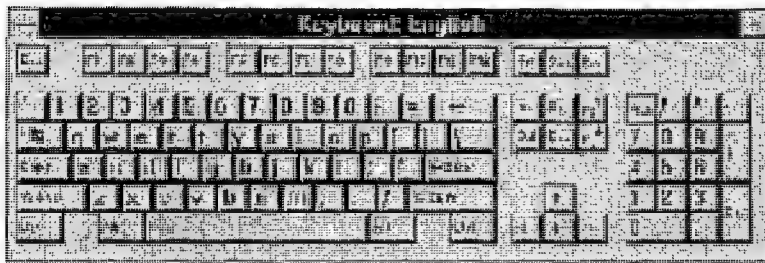
إن المصطلحات المستخدمة والعمليات التي تقوم بها الفأرة هي :

- الزر النشط : هو الزر الأيسر للفأرة .
- مؤشر الفأرة : وهو سهم صغير يتحرك على الشاشة عند تحريك الفأرة في أي اتجاه نريد ، ونستفيد منه في التأشير على مكان معين على الشاشة ، تدعى هذه العملية بالتأشير Pointing .

- النقر Clicking : ضغط الزر الأيسر للفأرة مرة واحدة و إفلاته بسرعة دون تحريك الفأرة على رمز أو أمر أو ايقونه معينة يقع عليها مؤشر الماوس وبالتالي تستخدم عملية النقر للتحفيز .
- النقر المزدوج Double Clicking يعني ضغط الزر الأيسر للفأرة مرتين متتاليتين على المكان الذي تقع عليه مؤشر الفأرة ، وتستخدم هذه العملية للتنفيذ .
- النقر والسحب Dragging يعني مواصلة الضغط على زر الفأرة الأيسر وتحريك الفأرة إلى مكان آخر من الشاشة ثم إفلات زر الفأرة ، ويستخدم من اجل نقل ايقونة أو نافذة معينة إلى مكان آخر على الشاشة .

ثالثاً : استعمال لوحة المفاتيح Using Keyboard

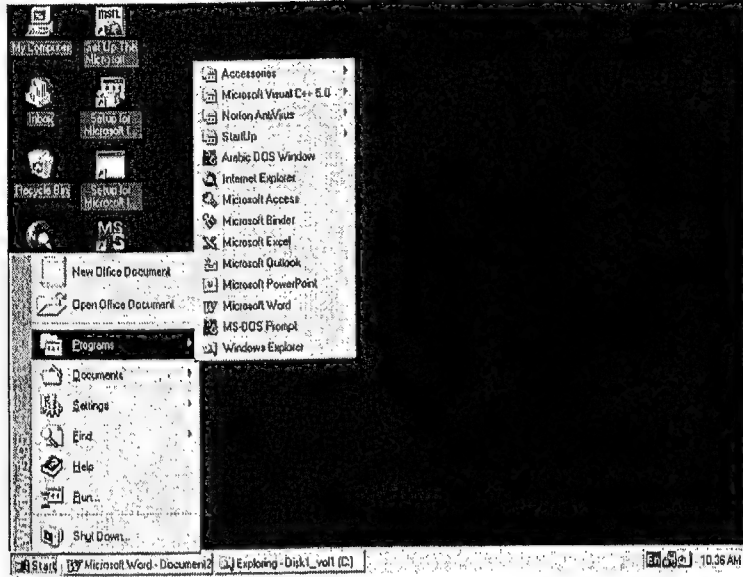
كما هو معروف فإن لوحة المفاتيح تعتبر من أهم وحدات إدخال البيانات إلى الذاكرة و تتضمن الحروف الأبجدية (العربية والإنجليزية) و حروف الترقيم والأرقام ،بالإضافة إلى مفاتيح خاصة تدعى مفاتيح وظيفية (Function Keys) ، أنظر الشكل (7) ، تقوم بمهام حسب البرنامج التطبيقي ، فمثلاً مفتاح Shift مع أحد مفاتيح الأحرف يؤدي إلى طباعة الحرف العلوي من المفتاح.



الشكل (7) لوحة المفاتيح

رابعاً : استعمال زر " ابدأ " : Using the Start Button

يظهر زر البدء Start في اسفل نافذة التشغيل على شريط المهمة Task Bar أنظر الشكل (5) ، عند النقر عليه تظهر قائمة البدء التي تضم مجموعة من الأوامر التالية (الشكل (8)) :



الشكل (8) قائمة زر البدء start

أمر البرامج Programs : بالنقر عليه تظهر قائمة فرعية تبين التطبيقات المخزنة داخل الجهاز ويمكن تشغيل أي من التطبيقات بالنقر المزدوج عليها من هذه البرامج برامج وورد (Microsoft Word) ، برنامج اكسل (Microsoft Excel) برنامج فوكس برو (Microsoft FoxPro)... ملاحظة : أن وجود الإشارة أمام الأمر تعني أنه هناك قائمة فرعية تظهر عند اختيار هذا الأمر .

وجود ثلاثة نقاط أمام الأمر ... يعني أنه يظهر عند اختيار هذا الأمر صندوق حوار Dialog Box يعطي إمكانيات إضافية للتحكم والسيطرة على تنفيذ الأمر المختار .

- أمر مستندات Documents : بالنقر عليه تظهر قائمة فرعية تبين المستندات (ملفات) تم العمل بها مسبقا ، ويتم فتح أي ملف بالنقر المزدوج عليه.
- أمر إعدادات Setting : بالنقر عليه تظهر قائمة فرعية تظم ثلاثة أوامر:
 1. لوحة التحكم Control Panel : وظيفتها التحكم في اعداد الأيقونات، وتغيير اللغة والتحكم في سرعة النقر بالفأرة وشكل المؤشر وأزرار الفأرة بالإضافة الى تغيير الرقم السري Password والتحكم بألوان عناصر النوافذ،...
 2. اعداد الطابعة Printers : وظيفته فحص حالة الطابعة وتوقيف الطابعة بشكل مؤقت واستئناف الطابعة ، ...
 3. شريط المهام Task Bar : لتحديد من خلاله بعض الخصائص المتعلقة بشريط المهام من إخفاء أو تصغير أو تكبير أو تغيير مكانه أو أية خصائص اخرى ممكنة .
- أمر بحث Find : بالنقر عليه تظهر قائمة فرعية تظم أمرين :
 - Files or Folder : مهمته البحث عن ملف معين أو البحث عن مجلد معين Folder .
 - Clean Sweep
- أمر Help : بالنقر عليه يظهر ملف التعليمات المساعدة لارشاد المستخدم عن كيفية استخدام أمر معين أو تنفيذ مهمة معينة من مهام نظام ويندوز .
- أمر Run : لتشغيل برنامج أو ملف معين .
- الأمر Shut Down : بالنقر عليه يؤدي الى إنهاء العمل على ويندوز وهنا يجب أن تكون جميع النوافذ مغلقة قبل استخدام أمر Shut Down .

خامساً : التحكم بالنافذة Controlling a window

يظهر النظام ويندوز 95 كل معلوماته في داخل صناديق أطر على الشاشة تدعى بالنوافذ Windows ، وقبل أن تتمكن من التعامل مع معلومات أي نافذة لا بد من تعلم كيفية فتحها ، تصغيرها وتكبيرها ، نقلها إلى أي مكان على الشاشة وتغيير حجمها .


أ. فتح نافذة : Opening a window

- نضع مؤشر الفأرة على رمز " جهاز الكمبيوتر My Computer "
- نقر زر الفأرة الأيمن ، تظهر القائمة السريعة .
- نختار الأمر " فتح " Open .
- تظهر نافذة شبيهة بالنافذة الرئيسية باسم My Computer تتألف من :
- 1. شريط العنوان : Title Bar : لإظهار اسم النافذة المفتوحة .

2. في الطرف الأيسر من شريط العنوان يوجد زر قائمة التحكم Control Menu يعطي للمستخدم إمكانيات التحكم بالنافذة المفتوحة ، عند النقر عليها بواسطة زر الفأرة الأيسر ، باستخدام أوامر مثل تصغير ، أو تكبير ، أو تحريك النوافذ . إذا نقرنا على زر قائمة التحكم بواسطة زر الفأرة الأيمن تظهر القائمة السريعة التي تنظم أيضاً بعض الأوامر المفيدة .

3. الطرف الأيمن من شريط العنوان يوجد ثلاثة أزرار كل واحد يؤدي مهمته معينة عند النقر عليه بواسطة زر الفأرة الأيسر :

تصغير نافذة الى زر يظهر على شريط المهام .  Minimize Button

يمكن تكبير النافذة بحيث تغطي كل الشاشة .  Maximize Button

إعادة النافذة إلى حجمها الأصلي، ويظهر هذا
الزر عندما تكون النافذة في حالة الحجم
الاقصى أي تغطي كامل الشاشة .



Restore Button

إغلاق النافذة .



Close Button

4. تحت شريط العنوان يظهر شريط القوائم Menu Bar والذي يضم أسماء
عدة قوائم وكل قائمة، عند النقر عليها بزر الماوس الأيسر ، تضم مجموعة
من الأوامر .

5. شريط الأدوات Tool Bar ويضم أزرار تمثل أوامر مهمات معينة بهدف
توفير الوقت للمستخدم بدلاً من الأوامر الموجودة داخل القوائم .

6. شرائط التصفح Scroll Bar ، وتظهر أسفل الشاشة أو على يسارها
مرسوم عليها أسهم وظيفتها إظهار القسم المخفي من محتويات النافذة عند
النقر على أحد الأسهم ، وهنا تشير أن شريط التصفح يظهر فقط عندما
يكون حجم محتويات النافذة أكبر من مساحة النافذة المفتوحة .

7. شريط المهام Task Bar ويظهر في أسفل النافذة ويمكن أن يضم ميقاتيّة
الوقت على الجهة اليمنى وزر ابدأ (Start) على الجهة اليسرى بالإضافة
إلى اسهم النافذة المفتوحة أو الأوامر المشار إليها بواسطة الماوس أو المهمة التي
يقوم بها الحاسوب .

8. الرمز (الأيقونات) Icon وهي رسوم صغيرة لتطبيقات مصغرة إلى الحد
الأدنى ويتم فتحها بالنقر المزدوج عليها .

ب. نقل النافذة وتغيير حجمها Moving and Resizing

1. نقل النافذة : نضع مؤشر الفأرة على شريط العنوان Title Bar ثم نضغط
على زر الفأرة الأيسر مع مواصلة الضغط والسحب إلى موقع جديد ، عندئذ

سيتحرك إطار النافذة مع حركة مؤشر الفأرة ، وعند تحرير زر الفأرة تستقر النافذة مع محتوياتها في مكانها الجديد .

2. تغيير حجم النافذة : نضع مؤشر الفأرة على أحد أضلاع (إطار) النافذة عندئذ يتغير شكل مؤشر الفأرة ويصبح على شكل سهمين متعارضين \longleftrightarrow ، نضغط على زر الفأرة الأيسر مع السحب إلى اليمين أو اليسار حسب الحجم الذي نريد .

ويمكن أيضا بنفس الطريقة سحب إحدى زوايا النافذة لتحريك ضلعين في نفس الوقت سواء للتكبير أو التصغير .

سادساً : ترتيب النوافذ والرموز على سطح المكتب

Arranging Windows and Icons on the desktop

إن ترتيب النوافذ على سطح المكتب يفيد في تنظيم العمل ، ويعطي رؤية واضحة للعناصر لا سيما أن كثرة النوافذ على سطح المكتب تجعل من الصعب إيجاد ما نبحث عنه بسهولة . لذلك ولتقدم فعالية عمل أكبر في ويندوز 95 نتبع مايلي :

1. نضع مؤشر الفأرة على شريط المهام Task Bar

2. ننقر زر الفأرة الأيمن فتظهر قائمة تظم الأوامر التالية :

- | | |
|----------------------|-----------------------|
| Cascade | - تنالي |
| Tile Horizontally | - بجانب أفقي |
| Minimize All Windows | - تصغير كافة الإطارات |
| Tile Vertically | - بجانب عمودي |
| Undo Tile | - تراجع تنالي |
| Properties | - خصائص |

نختار أحد الأوامر السابقة حسب ما نريد ترتيب النوافذ المفتوحة على الشاشة.

أما لترتيب الرموز المبعثرة على سطح المكتب فيتم بالشكل التالي :

1. نقر زر الفأرة الأيمن على أي منطقة خالية من سطح المكتب .

2. تظهر قائمة أحد أوامرها ترتيب الرموز Auto Arrange

3. نختار الأمر ترتيب الرموز فيتم ترتيب الايقونات بشكل تلقائي

سابعاً : استعمال القوائم Using Menus

معظم النوافذ في ويندوز 95 يحتوي على شريط القوائم Menus Bar

الذي يضم مجموعة من أسماء القوائم مثل :

ملف File ، تحرير Edit ، عرض View ، خيارات Option ، تعليمات

Help ، وغير ذلك حسب التطبيق المفتوح .

وكل قائمة من القوائم تضم مجموعة من الأوامر كل منها له مهمة معينة ، وهنا

يجب أن نشير إلى بعض المصطلحات التي يمكن أن نشاهدها عند التعامل مع القوائم



والتي ندرجها في الجدول التالي :

المعنى	ما نشاهده في القائمة
عند اختيار الأمر سيظهر صندوق حوار يحوي المزيد من الخيارات .	1. ثلاثة نقاط بعد الأمر (...)
لا يمكن تنفيذه في الوقت الحالي لأسباب تتعلق بطبيعة الأمر نفسه	2. أمر لون كاشف (إضاءة خافته)
عند اختيار الأمر سيظهر قائمة ثانوية أخرى تحوي المزيد من الأوامر	3. مثلث بعد الأمر على شكل سهم
عند ظهورها قبل الأمر تعني أن الأمر فعال ويمكن لأكثر من أمر في المجموعة الواحدة أن يحمل علامة التحقق ويتم إظهارها بالنقر بزر الماوس الأيسر على الأمر، أما إخفاءها فيتم بالنقر مرة أخرى على الأمر	4. علامة التحقق قبل الأمر (check mark)
تعني أن الأمر فعال ، ولا يمكن لأكثر من أمر في المجموعة الواحدة أن يحمل نقطة ولوضع النقطة نقر زر الماوس الأيسر على الأمر ، أما إخفاءها	5. نقطة بلون غامق قبل الأمر (•)

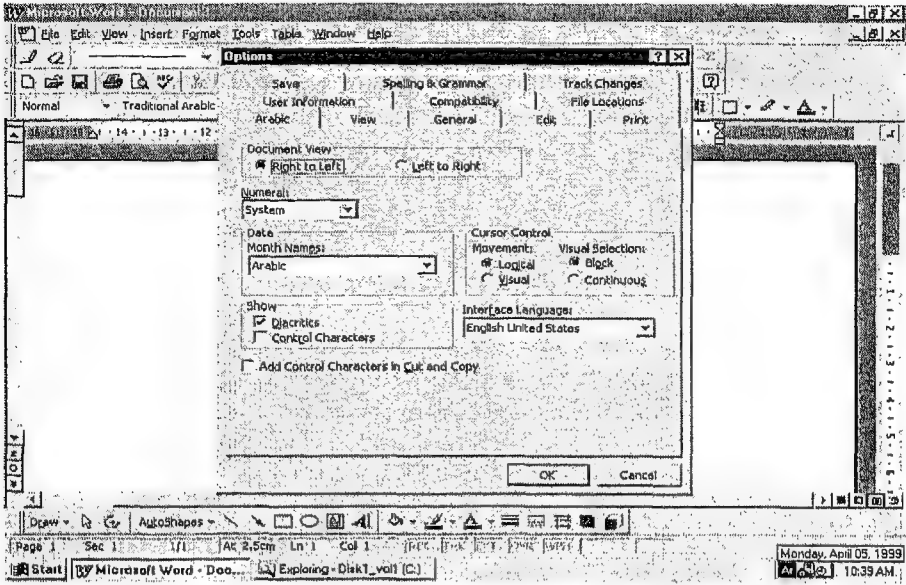
فيتم بالنقر على أمر آخر ضمن المجموعة.	
يمكن تنفيذ الأمر بواسطة المفتاح المشار اليه مثل ctrl+p للطباعة بدلا من فتح القائمة في كل مرة.	6. اسم مفتاح ائتلاف بعد الأمر

ثامناً : استعمال صناديق الحوار Using a Dialog Box

صندوق الحوار Dialog Box يظهر بشكل مستطيل ، يحتوي على عدد من العناصر والخصائص الإضافية التي يمكن اختيار بعضها أو كلها حسب ما تتطلب الحاجة ، وأغلب الصناديق في النظام التشغيل ويندوز تحمل عناصر مشتركة يتم التعامل معها بطرق متشابهة ، أنظر الشكل (9) والتي ندرجها في الجدول التالي:

اسم العنصر	شكل العنصر	الشرح
1.مفتاح الخيارات Options buttons		مفاتيح دائرية بيضاء يؤدي اختيارها بالنقر عليها بزر الفأرة الأيسر الى ظهور نقطة سوداء داخل الدائرة البيضاء، وهذا يعني أن الخيار فعال وهنا يجب أن نلاحظ أنه يمكن تفعيل خيار واحد فقط من ضمن كل مجموعة من مجموعات صندوق الحوار .
2.صناديق التحقق Check Boxes		مربعات صغيرة أمام الخيارات، يتم تفعيل الخيار بالنقر بزر الفأرة الأيسر مرة واحدة عليه وتظهر إشارة التحقق داخل المربع ولإزالة التفعيل ننقر مرة أخرى على الخيار. ملاحظة: يمكن تفعيل أكثر من مربع ضمن المجموعة الواحدة على عكس مفاتيح الخيارات .

<p>وهي لائحة تحتوي على مجموعة من العناصر (مثل الملفات) والتي يمكن اختيار إحداها ولفتح اللائحة نقر بزر الفأرة الأيسر على السهم الموجود على يمين اللائحة، ويمكن تصفح محتويات اللائحة بتدويرها الى الاعلى أو الاسفل باستخدام الأسهم التي تظهر على أحد جانبي اللائحة</p>		<p>3. صناديق اللوائح المنسدلة List Boxes</p>
<p>صناديق يمكن إدخال مقاييس فيها ، مثل إدخال حجم الخط ، عدد النسخ، اسم الملف ...</p>		<p>4. الصناديق النصية Text Boxes</p>
<p>وهو شكل مربع أو مستطيل مكتوب عليه: ok أي الموافقة على الخصائص والخيارات المختارة cancel إلغاء الأمر والإبقاء على الخيارات القديمة ويمكن أن يكون مكتوب عليها أوامر أخرى تؤدي عند اختيارها إلى صندوق حوار آخر أو إنجاز مهمة معينة مثل زر Options</p>		<p>5. أزرار الأوامر Command Buttons</p>
<p>عبارة عن صفحات ضمن صندوق الحوار تحتوي على معلومات وخيارات محددة وعادة تكون هذه الألواح في أعلى الصندوق واللوح الفعال يكون بشكل مستطيل نافر ، لاختيار صندوق آخر نقر بزر الفأرة الأيسر على أسم اللوح الآخر فنجد أن الخيارات قد تغيرت بخيارات أخرى تناسب اللوح المفتوح.</p>		<p>6. الألواح Tabs</p>



الشكل (9) أحد نماذج صناديق الحوار

(3-5-5) التعامل مع المجلدات والملفات

Working With Folders and Files

(1-3-5-5) مقدمة :

لقد تطرقنا في الفقرة (4-5) إلى نظام التشغيل MS-DOS وتعرفنا على وظائفه ومهامه وكيفية التعامل مع الملفات والفهارس وإجراء العمليات الضرورية عليها ، أما في هذه الفقرة فسنتعرف على كيفية التعامل مع الملفات والمجلدات (الفهارس) ضمن نظام تشغيل مطور وهو نظام التشغيل ويندوز 95 (WINDOWS 95) ، الذي يسهل الكثير من العمليات وقبل أن ندخل بالتفاصيل سنقدم لمحة بسيطة عن MS-DOS نلخص من خلالها أهم المهمات والوظائف لهذا النظام وعلاقة هذه الوظائف بنظام التشغيل ويندوز .

يتألف كل قرص مرن أم صلب أم مضغوط CD من فهرس رئيسي توضع عليه الملفات والفهارس الفرعية ، وتتألف الفهارس الفرعية بدورها من

فهارس فرعية أخرى وملفات ، والفرق بين الملف والفهرس هو أن الفهرس شكل تنظيمي يتم تجميع فيه الملفات والفهارس الفرعية المنتمية الى موضوع محدد أي مكان تخزيني شبيه بقرص ، أما الملف فهو عبارة عن مجموعة من البيانات ذات علاقة معينة تخزن تحت اسم معين .

يمكن التمييز بين العمليات التي تنجز على الفهارس وبين التي تجري على الملفات حيث نجد أن أهم عملية يتعرض لها المستخدم عند العمل على ملفات لبرامج هي تشغيل البرنامج بكتابة أسم الملف عند محث نظام التشغيل MS-DOS ثم الضغط على ENTER وأهم عملية تنجز على الملفات النصية هي فتح هذه الملفات بواسطة محررات النصوص لتعديلها أو الإضافة عليها أو قراءتها... الخ ، أما بالنسبة للفهارس فإن أهم عملية هي الدخول إلى الفهارس للوصول الى محتوياته باستخدام الأمر (CD) يليه أسم الفهرس الفرعي ، ثم نضغط ENTER بالإضافة الى الأوامر الأخرى التي تساعد على نقل الملفات من فهرس الى فهرس آخر أو نسخها أو حذفها أو ... الخ ، وكل هذه الأوامر يجب حفظها ليتم إدخالها عند محث النظام DOS ثم الضغط على مفتاح الإدخال ENTER.

مع ظهور أنظمة التشغيل ويندوز ، أصبح التعامل مع تلك الأوامر اسهل وأسرع حتى أن الكثير من مستخدمي الحاسوب يفضلون التعامل مع نظام التشغيل ويندوز واعتبروا أن الدوس طراز قديم وعقيم الفهم والتعامل .

لقد أصبح الفهرس الفرعي Subdirectory في نظام ويندوز 95 يدعى مجلدًا Folder ، ويتم الانتقال إليه بالنقر عليه نقرة واحدة بواسطة الفأرة، أما فتحه واستعراض محتوياته فيكون بالنقر المزدوج عليه ، كما سنشرح ذلك لاحقاً، أما بالنسبة للملفات فيؤدي النقر المزدوج بواسطة الفأرة على ملف تنفيذي لتطبيق معين الى تشغيل ذلك التطبيق ، بينما يؤدي النقر المزدوج على ملف نصي الى جلب التطبيق الذي كتب فيه ذلك المستند وفتحه . بالإضافة الى ما سبق فإن

ويندوز 95 يقدم خدمات أخرى تساعد على تنظيم العمل على المجلدات والملفات مثل خاصية البحث عن ملف أو مجلد وتحديد موقعه والانتقال إليه .

(5-5-3-2) طرق التعامل مع الملفات والفهارس :

يقدم النظام ويندوز 95 طريقتين للتعامل مع الفهارس والملفات :

الطريقة الأولى : وتمثل باستخدام ايقونة "جهاز الكمبيوتر" "My Computer" الموجودة على سطح المكتب كما يلي :

1. النقر المزدوج على ايقونة جهاز الكمبيوتر "My Computer" تظهر نافذة تحوي ايقونات مختلفة بالإضافة إلى ايقونات محركات الأقراص (سواقات الأقراص) المتوفرة على الحاسوب مثل A,B , ومحرك الأقراص المضغوطة .

2. نقوم بالنقر المزدوج على أحد رموز محركات الأقراص ، يؤدي ذلك الى ظهور نافذة تستعرض محتويات ذلك الحرك ، وتسمح بإجراء العمليات اللازمة فالتنسخ والنقل والحذف وغير ذلك .

الطريقة الثانية : فتمثل باستخدام ، مستكشف ويندوز "Windows Explorer" كما سنشرح في الفقرة التالية :

الاختلاف بين الطريقتين الأولى والثانية هو أن استخدام مستكشف يسمح بمشاهدة جميع الملفات والمجلدات على الفهرس الرئيسي ضمن القسم اليساري لنافذة المستكشف ، بينما يعرض في القسم الأيمن من النافذة محتويات المجلد المختار ، بينما يؤدي استخدام طريقة "جهاز الكمبيوتر" إلى فتح نافذة جديدة لعرض محتويات المجلد مما يتسبب بامتلاء الشاشة بالنوافذ ، بالإضافة إلى أن مستكشف النوافذ هو عبارة عن مدير الملفات في ويندوز 3.X ، بل يعتبر نسخة مطورة عنه .

(5-5-3-3) : استخدام مكتشف النوافذ

Using The Windows Explorer

مستكشف النوافذ هو أداة لتنظيم الملفات والفهارس داخل الحاسوب ، حيث يمكننا من عرض محتويات الأقراص والفهارس ، ونقل ونسخ وحذف الملفات وهيئة الأقراص ، وإنشاء فهارس جديدة وإعادة تسمية الفهارس وتشغيل التطبيقات الخ...

إن العمل على مستكشف النوافذ في ويندوز 95 أو ويندوز 98 يؤدي لنفس الغرض والمهام التي نقوم بها باستخدام نظام التشغيل MS-DOS ، بالإضافة إلى أسلوب التعامل بين المستخدم والحاسوب أفضل وأسهل لذلك سنقدم في الفقرة القادمة بعض أوامر الدوس بأسلوب ويندوز 95 ، بما يتناسب مع مناهج هذه المادة ، وبما يلي حاجة القارئ في البدء للعمل على نظام التشغيل ويندوز 95 أو 98 .

1. تشغيل مستكشف النوافذ :

- النقر على زر ابدأ (START) ، تظهر قائمة ابدأ .
- النقر على أمر البرامج Programs ، تظهر قائمة فرعية بالملفات والبرامج .
- النقر على Windows Explorer مستكشف النوافذ ، عندئذ تظهر النافذة المطلوبة كما في الشكل (10) .

تألف نافذة مستكشف النوافذ كما في أي نافذة من :

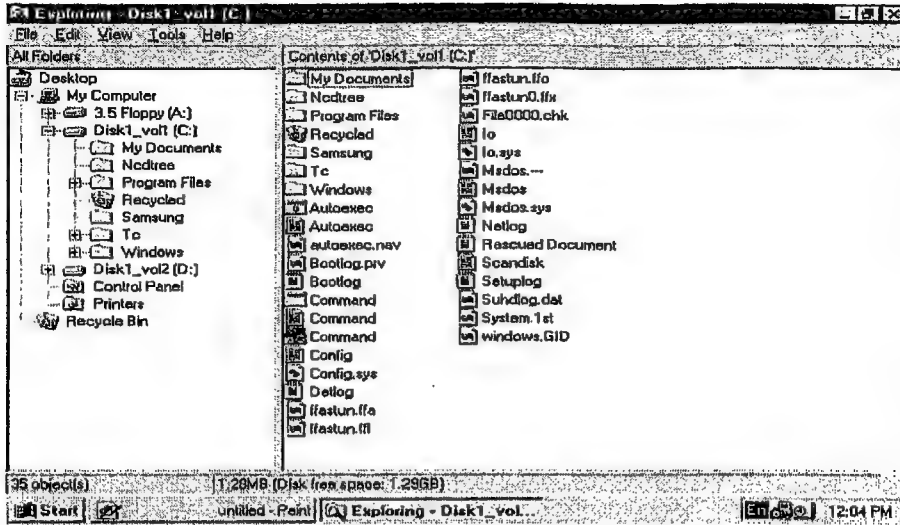
1. شريط العنوان Title Bar والذي يحوي في الطرف الأيمن ثلاثة أزرار :

(تصغير □ ، تكبير □ ، إغلاق ✕)

2. شريط القوائم Menu Bar ويظم القوائم :

File الملف

Edit تحرير



الشكل (10) نافذة مستكشف النوافذ

3. شريط الأدوات : الذي يضم أزرار تستخدم اختصارا بدل أوامر موجودة في القوائم مثل القص، النسخ ، اللصق ، بالإضافة الى صندوق يظهر فيه الفهرس (المجلد) المختار .

4. سطح المكتب أو المنطقة المعروض فيها الفهارس وتقسم إلى قسمين الأيسر يضم جميع المجلدات (الفهارس) All Folders بما أنهما محرك الأقراص ، والقسم الأيمن يضم الملفات والفهارس الفرعية الموجودة داخل المجلد (الفهرس) المختار .

2 - فتح نافذة فهرس جديد بواسطة رمز محرك الأقراص :

بالنقر المزدوج على رمز محرك الأقراص (A) تظهر جميع الملفات والفهارس الموجودة في محرك الأقراص المختار تحت العنوان (Name) في القسم الأيمن من النافذة وعندها يمكن فتح أي ملف بالنقر المزدوج على الملف المطلوب .

3- تهيئة الأقراص المرنة Formatting

ويعني تجهيزها وفحصها من الأعطال قبل التخزين عليها أما إذا كان القرص يحوي معلومات مخزنة فإن عملية التهيئة تؤدي الى حذف جميع المعلومات على ذلك القرص ، ولتهيئة القرص المرن نتبع ما يلي :

1. نضع القرص المرن في محرك الأقراص المرنة (A).
2. نقر بزر الفأرة الأيسر على رمز محرك الأقراص (A).
3. نقر بزر الفأرة الأيمن ، فتظهر قائمة تحوي مجموعة من الأوامر .
4. نقر الأمر Format عندئذ يظهر صندوق حوار باسم Format .
5. من صندوق القائمة المنسدلة (Capacity) نختار حجم القرص المراد تهيئته نختار الحجم 1.44 MB .
6. نقر على الزر Full ليتم شطب محتويات القرص كلياً .
7. نقر على زر Start ليبدأ الحاسوب بتهيئة القرص .

4- ترتيب الملفات Arrange Files

1. نقر زر الفأرة الأيمن على القسم الأيمن من النافذة فتظهر قائمة جديدة تضم مجموعة من الأوامر .
2. نختار الأمر Arrange Icons ، عندئذ تظهر قائمة فرعية جديدة تحوي عدة أوامر .
3. لفرز المجلدات والملفات حسب الترتيب الأبجدي للمحقق الاسم (اللاحقة) نختار الأمر "By Type" الفرز حسب النوع .
4. لفرز المجلدات والملفات حسب حجمها تنازلياً نختار الأمر "By Size" الفرز حسب الحجم .
5. لفرز المجلدات والملفات حسب تاريخ إنشائها بدأً من التاريخ الأخير "By Date" الفرز حسب التاريخ .

5 - تحديد (اختيار) مجلد أو ملف **Selecting Files Or Folders**

ويتم ذلك بالنقر بزر الفأرة الأيسر على أسم الملف أو المجلد المراد اختياره وفي حالة تحديد أكثر من ملف عندئذ نضغط على المفتاح Ctrl و زر الفأرة الأيسر على الملفات المراد تحديدها .

6 - إنشاء فهرس (مجلد) **Create Folder**

- لإنشاء مجلد جديد داخل مجلد آخر موجود نتبع ما يلي :
1. تحديد المجلد المراد إنشاء مجلد فرعي داخله .
 2. نفتح القائمة File من شريط القوائم .
 3. نختار الأمر New ، تظهر قائمة فرعية .
 4. نختار الأمر Folder فيظهر صندوق حوار .
 5. ندخل الأسم الذي نريده للفهرس ضمن الحقل المخصص ثم نضغط على زر ok

7- إعادة التسمية **Renaming** :

- من أجل تغيير اسم أحد المجلدات أو الملفات نتبع ما يلي :
1. تحديد الملف أو المجلد المراد تغيير اسمه .
 2. نفتح القائمة File من شريط القوائم .
 3. نختار الأمر Rename عندئذ يظلل الأسم المراد إعادة تسميته .
 4. نطبع الأسم الجديد ثم نضغط مفتاح الإدخال Enter من لوحة المفاتيح.

8- حذف ملف أو مجلد **Delete File Or Folder** :

- لحذف ملف أو مجلد نتبع ما يلي :
1. نحدد الملف أو المجلد المراد حذفه .
 2. نفتح القائمة File من شريط القوائم .

3. نختار الأمر Delete عندئذ تظهر رسالة لتأكيد الحذف .
4. نختار الزر (ok) في حالة حذف ملف واحد أو مجلد واحد والزر (Yes To All) في حالة حذف عدة ملفات أو مجلدات .

9- نسخ الملفات أو المجلدات Copying Files Or Folders :

1. نحدد الملف أو المجلد المراد نسخه (عمل نسخة منه) .
2. نفتح القائمة Eidet من شريط القوائم .
3. نختار الأمر Copy .
4. ننقر عند المكان الذي نريد وضع النسخة من الملف أو المجلد فيه .
5. نضغط مفتاح الإدخال Enter من لوحة المفاتيح .

10- نقل الملفات أو المجلدات Moving Files Or Folders :

1. نحدد الملف أو المجلد المراد نقله .
2. نفتح القائمة Edit من شريط القوائم .
3. نختار الأمر Cut .
4. ننقر عند المكان الذي نريد وضع الملف أو المجلد المنقول .
5. نضغط مفتاح الإدخال Enter من لوحة المفاتيح .

11- إغلاق مكتشف النوافذ Ending Explorer :

وهنا لدينا أسلوبين :

1. نختار الأمر Close من القائمة File .
2. النقر على زر الإغلاق الموجود في الجهة اليمنى من شريط العنوان .

(5-6) البرمجيات التطبيقية الجاهزة : PACKAGES

هناك العديد من البرامج التطبيقية الجاهزة كل منها يلي احتياجات معينة للمستخدم حسب مجال عمله واختصاصه ، ولا تحتاج هذه البرامج من المستخدم معرفة متعمقة بعلم الحاسوب أو حتى تركيب الحاسوب ومكوناته المادية ، بل

يكفي تعلم المستخدم كيفية التعامل مع البرنامج بحد ذاته وكيفية الدخول إليه واستخدام ميزاته المختلفة .

سنقدم في هذه الفقرة تعريف ببعض هذه البرامج الشائعة الاستخدام لدى مختلف المستويات الاجتماعية ، ونترك التفاصيل الجزئية للعمل على هذه البرامج واستثمارها لمادة برمجيات الحاسوب الشخصي (1) وبرمجيات الحاسوب الشخصي (2) والإحصاء المحوسب .

(1-6-5) برامج معالجة النصوص Word Processors

تعتبر برامج معالجة النصوص من برامج الحاسوب التطبيقية الأكثر انتشاراً والهامّة في تكنولوجيا المعلومات ، والمقصود من معالجة النصوص هو تسخير مكونات الحاسوب وبرمجياته في إدخال نص كتابي الى نظام الحاسوب ، وتخزينه ، التعديل عليه، عرضه على الشاشة بأشكال مختلفة ، استرجاعه وقت الحاجة ، تنسيقه بالشكل المناسب ، طباعته ، وهناك العديد من برامج معالجة النصوص مثل : Word Perfect , Works , Write , Microsoft Word ... وغيرها .
وبرنامج مايكروسوفت وورد Microsoft Word أكثر هذه البرامج شيوعاً ولهذا سوف نتناول بعض ميزاته :

أولاً : يتم تشغيله من داخل نظام التشغيل ويندوز ، القائمة Start الموجودة أسفل الشاشة ثم النقر على الأمر برامج Program ، ثم النقر على Microsoft Word عندئذ تظهر نافذة الورد انظر الشكل (11) .

ثانياً : العمليات الرئيسة لبرنامج مايكروسوفت وورد :

1. إنشاء نص Creating :

يتم إنشاء نص باستخدام لوحة المفاتيح Keyboard ، بعد النقر على الأمر New من القائمة File الذي يعطي إمكانية إنشاء مستند

(صفحة للكتابة) يظهر فيها مؤشر صغير Cursor في بدايتها ، نقوم بإدخال الكلمات من لوحة المفاتيح عند المؤشر ويتم الانتقال إلى السطر التالي تلقائياً عند وصول المؤشر إلى نهاية السطر دون الحاجة إلى الضغط على مفتاح الإدخال Enter، وبعد الانتهاء من الكتابة يمكن استخدام الأمر تخزين Save من القائمة File لتخزين الملف على أحد وسائط التخزين المساعدة باسم معين.

2. التحكم بالهوامش Indentation :

برنامج الورد يعطي إمكانية التحكم بالهوامش اليمنى واليسرى ورأس الصفحة واسفل الصفحة حسب ما تقتضي الحاجة وحسب نوع الورق الذي سيستخدم للطباعة ، بالإضافة الى إمكانية التحكم بالتباعد بين الفقرات Spacing والتباعد بين الأسطر line Spacing.

3. التعديل على النص المخزن Editing :

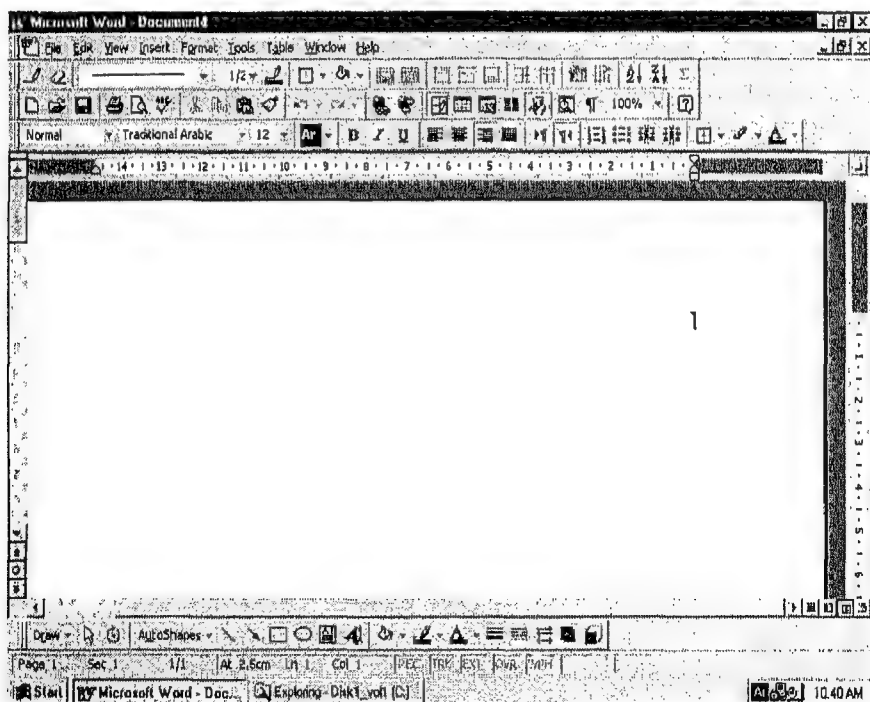
تعتبر إمكانية التعديل على النص من أقوى ميزات برنامج الورد التي تجعله يتفوق على وسائط الكتابة التقليدية مثل الآلات الكاتبة ، رغم أن التعديل متاحاً أثناء الكتابة ، إلا أن المستخدم يستطيع أيضاً الرجوع الى النص الذي انتهى من إدخاله في أي وقت وعرضه على الشاشة وإجراء التعديلات اللازمة ، وهذا يتيح للمستخدم مراجعة النص عدة مرات والتأكد من عدم وجود أخطاء قبل الطباعة . بالإضافة إلى ميزات أخرى يمكن استخدامها وهي نقل أو نسخ أي مقطع من النص الى مكان آخر في نفس المستند أو إلى مستند آخر.

4. التحكم بحجم ونوع الخط Controlling Font Kind & Size

يتوفر في الورد أنواع مختلفة من الخطوط العربية والأجنبية ، الأمر الذي يتيح للمستخدم تغيير النص المكتوب أو المدخل بأي نوع من الخطوط بالإضافة الى إمكانية تغيير حجم الخط في أي مكان من النص .

5. التدقيق الإملائي للنص : Spelling

حيث يقوم الحاسوب باستخدام خاصية التدقيق الإملائي بمقارنة كل كلمة من النص بالكلمات المخزنة في قاموس الورد Dictionary ، وإذا وجدت كلمة مختلفة عن نظيرتها في القاموس يقدم الورد عدة خيارات لتصحيحها ، إذ أن قاموس الورد يحتوي على عشرات الألف من الكلمات الشائعة الاستخدام ، مع إمكانية إضافة كلمات جديدة إليه .



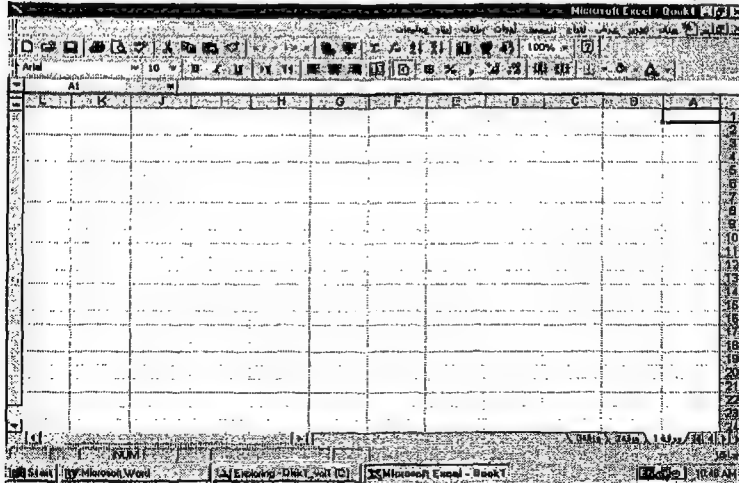
الشكل (11) نافذة مايكروسوفت وورد Microsoft Word

(5-6-2) الجداول الإلكترونية Spread Sheets

الجدول الإلكتروني هو جدول مؤلف من عدد كبير من الأعمدة والصفوف، تعرض الشاشة جزءاً صغيراً منه ، إلا أن المستخدم يمكنه التعامل مع الجزء المخفي عن طريق أزرار التصفح ، من وظائف الجداول الإلكترونية : تخزين البيانات والقيام بالعمليات الحسابية والتحليلات الإحصائية وإنشاء الرسوم البيانية لهذه البيانات باستخدام أوامر سهلة الاستعمال .

أول برامج الجداول الإلكترونية التي تم تصحيحها هو برنامج (Visual C) 1979 ثم ظهرت فيما بعد برامج كثيرة ومتطورة إلى أن أصبحت في الوقت الحالي تجمع بين خصائص الجداول الإلكترونية وخصائص برامج معالجة النصوص وبرامج إدارة قواعد البيانات من هذه البرامج ، برنامج (Ability Plus) وبرنامج (Louts 123) وبرنامج (Qwattropro 3) وبرنامج (Supercalc 5) وبرنامج (Excel) .

ويعتبر برنامج اكسل Excel أكثر البرامج استخداماً في المؤسسات والشركات التجارية بسبب سهولة العمل عليه وميزاته التي تلي حاجة المستخدم في المؤسسات والشركات .



الشكل (12) نافذة ميكروسوفت اكسل Microsoft Excel

أهم ميزات برنامج اكسل :

1. يتم تشغيله من داخل نظام التشغيل ويندوز ، القائمة Start الموجودة في اسفل الشاشة ثم النقر على الأمر برامج Program ، ثم النقر على X Microsoft Excel عندئذ تظهر نافذة ميكروسوفت اكسل انظر الشكل (12)

2. الاحتفاظ بالبيانات التي ندخلها الى ملفاته وإمكانية استرجاعها وقت الحاجة .
 3. سهولة التحكم بالبيانات من ناحية مظهرها وإنجاز العمليات الحسابية اللازمة عليها مثل الضرب - الجمع - الطرح - القسمة ، و العمليات الرياضية المعقدة مثل حساب المثلثات والدوال والعمليات الإحصائية والهندسية .
 4. إمكانية تحليل البيانات باستخدام الرسوم البيانية Graphs ، مع إمكانية التعديل على الرسوم وتلوينها وتكبيرها وتصغيرها ...
 5. إمكانية استخدام البرنامج Excel كقاعدة بيانات (أي تقسيم البيانات ضمن سجلات وحقول لتسهيل عمليات الفرز والاستخراج المشروط والبحث) .
 6. استخدام ميزة البرجة والماكرو أي إمكانية تحويل العمليات الروتينية الى عمليات تلقائية تنفذ كاملة بمجرد إنجاز خطوة واحدة .
- بالإضافة إلى ميزات كثيرة أخرى يمكن التعرف عليها عند التعلم على استخدامه في مادة الخزم البرمجيات (1) .

(3-6-5) برامج إدارة قواعد البيانات

Database Management Program

في عصرنا الحالي أصبحت المعلومات وسرعة وصولها الى متخذ القرار أيضاً كان موقعه عنصراً هاماً ومؤثراً على جوانب عديدة من المجتمع ، وذلك نتيجة التقدم العلمي والتطور التكنولوجي الذي رافقه فرزاً لحجم كبير من البيانات الأولية التي تحتاج الى نظم وأساليب للتعامل معها و معالجتها ، وقد لعب الحاسوب بإمكانيته الهائلة في إدارة وتنظيم هذه البيانات وصنع المعلومات ، سميت هذه النظم

به نظم إدارة قواعد البيانات (Database Management Programs) واختصاراً (DBMS). وسنقدم في هذه الفقرة شرحاً للمفاهيم الأساسية لقواعد البيانات :

(1-3-6-5) تعريف قواعد البيانات

يمكن تعريف قاعدة البيانات بأنها مجموعة من المعلومات والبيانات ذات صفات مشتركة والمخزنة بنظام بطريقة نموذجية تحقق نوعاً من الاستقلالية والمناعة ضد التغيير عن البرامج التي تقوم باستخدامها حاضراً أو مستقبلاً إذا اضطررنا لتغيير أي منها ، وتتألف قاعدة البيانات من مجموعة من السجلات مثلاً سجلات الموظفين أو سجلات الطلبة أو دليل الهاتف ، سجلات المرضى في مستشفى سجلات المسافرين وحركة الطائرات في شركات الطيران ، سجلات حسابات العملاء في بنك ويدير قاعدة البيانات عادة نظام يدعى بنظام إدارة قواعد البيانات. لذلك فإن نظام إدارة قواعد البيانات يتكون من قسمين :

القسم الأول : قاعدة البيانات والتي هي مجموعة من البيانات المخزنة على شكل ملفات مرتبطة مع بعضها البعض مثل حجوزات الطيران ، تسجيل الطلبة ، حسابات العملاء في بنك ...

القسم الثاني : برنامج إدارة قاعدة البيانات والتي هي عبارة عن مجموعة من البرامج التي تقوم بتشغيل ملفات البيانات المخزنة لتنفيذ العديد من العمليات مثل الإضافة والحذف والتحديث والبحث والاسترجاع الجزئي والكلي والتبويب والفرز ...

(2-3-6-5) مكونات بيئة نظم قواعد البيانات

تتألف بيئة نظم قواعد البيانات من خمسة مكونات أساسية كما يلي :

- 1- المكونات المادية : Hard ware وتتألف من الحاسوب وتوابعه .
- 2- المكونات البرمجية : Software وهي نظم خاصة بقواعد البيانات مثل

الترجمات ونظم التشغيل وبرمجيات الشبكات إذا كانت قواعد البيانات موزعة .

3- البيانات : Data وهي أهم مكونات نظم قواعد البيانات والتي من أجلها وجدت مثل هذه النظم .

4- الإجراءات : Procedures وهي التعليمات والقوانين التي تتحكم بقاعدة البيانات وكيفية الاستفادة منها والتحكم بالمخرجات .

5- المستخدمون : Users ويمكن تصنيف المستخدمين إلى أربعة مستويات:

1. مدير قواعد البيانات Database Administrator ، وهو شخص متخصص بقواعد البيانات وخبير لفترة طويلة في هذا المجال مهمته تحديد متطلبات قواعد البيانات من برمجيات وأجهزة والرقابة والضبط على أداء النظام ولديه الصلاحيات في التنسيق و التعامل مع الآخرين وحماية قاعدة البيانات من الاستخدام الخاطئ .

2. مصمم قاعدة البيانات Database Designer : ويكون إما شخص أو عدة أشخاص مهمتهم تحديد البيانات اللازم تخزينها ، تحديد أفضل تراكيب هياكل البيانات الواجب استخدامها ، تصميم واجهات وشاشات التخاطب بين المستخدم وقاعدة البيانات ، تصميم الهيكل التنظيمي الداخلي لتخزين البيانات وأساليب الوصول إليها .

3. مبرمجو قواعد البيانات Programmers ، مهمتهم برمجية قواعد البيانات وجعلها قابلة للتنفيذ في تطبيق محدد .

4. مستخدمو قواعد البيانات Users ، وهم الأشخاص المستفيدون من تطبيقات أنظمة قواعد البيانات مثل : موظف التسجيل ، المحاسب ، الإداري ،

(5-6-3-2) قواعد البيانات العلائقية Relational Databases

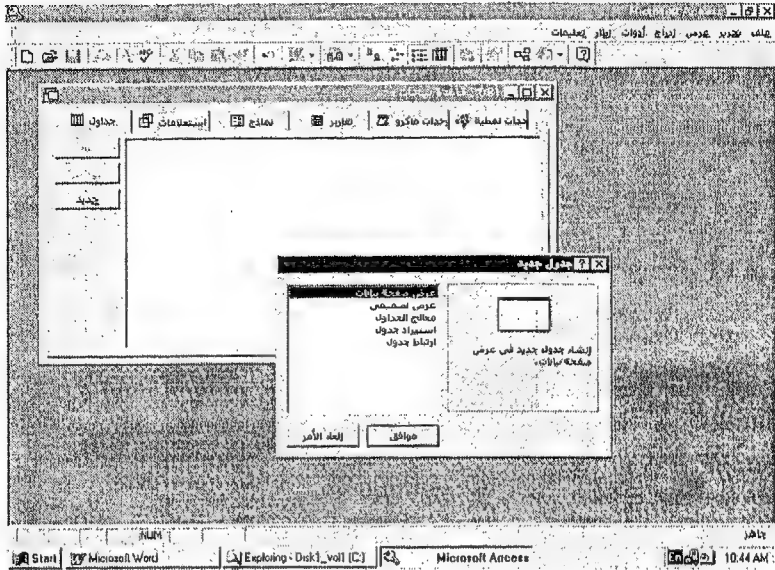
العلاقة هي الأساس مصطلح رياضي ، وتمثل بجدول ذو بعدين مكوناً من صفوف وأعمدة ، تشكل هذه الجداول نوعاً من وسائل تخزين البيانات وحفظها في قاعدة بيانات وقد تكون الجداول ذات علاقة تبادلية فيما بينها ، لذلك سميت بقواعد البيانات العلائقية ويعتبر عام 1970 البداية النظرية لنموذج قواعد بيانات علائقية والذي اقترحه العالم كود E. F. Codd وفي مطلع 1980 انتشرت هذه الفكرة وأصبحت أداة لتنفيذ التطبيقات التجارية .

وهناك العديد من نظم إدارة قواعد البيانات التي تتفاوت في كفاءتها حسب سرعة استدعاء البيانات فيها بالإضافة إلى مدى سهولة استخدامها .

من أشهر هذه النظم برامج أسرة (DBASE) مثل (DBASE II) , (FoxPro FoxBASE , DBASE IV , DBASE III , (Paradox) وبرنامج (Reflex) وبرنامج (Access) ...

ويعتبر برنامج أكسس أكثر برامج قواعد البيانات استخداماً وذلك لإمكاناته الضخمة ، الأمر الذي يسهل عملية إنشاء قاعدة البيانات وحفظها وتعديلها والبحث عن محتوياتها بأسرع وقت ممكن بالإضافة إلى إمكانية ربط جداول قواعد البيانات مع بعضها البعض ، كما أنه توجد إمكانية استخدام لغة البرمجة (Access Basic) ، وهذا يجد ذاته يدعم المبرمج ويوفر له إمكانيات برمجية ضخمة .

لا يمكننا هنا الخوض بتفاصيل الأكسس بل نتركه إلى مادة حزم برمجيات (2) إن شاء الله للتعرف على جزئياته وكيفية العمل عليه و الشكل (13) يبين الشاشة الرئيسية للميكروسوفت اكسس Microsoft Access وأقسامها .



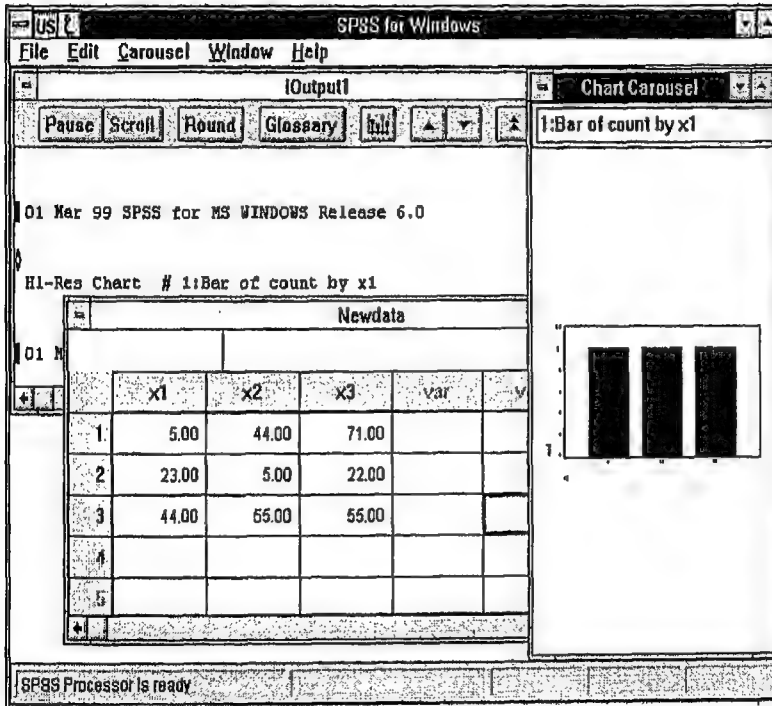
الشكل (13) الشاشة الرئيسية لمايكروسوفت اكسس Microsoft Access

(5-6-4) برامج الإحصاء الحوسبي

مع التطور العلمي والتكنولوجي أصبح من الصعب إجراء التحليلات الإحصائية بالطرق اليدوية نتيجة لتضخم حجم البيانات الأولية المراد دراستها ، بالإضافة الى أن أغلب الدراسات في مجالات العلوم الاقتصادية والتربوية والاجتماعية والسياسية تعتمد وبشكل أساسي على تحليل البيانات إحصائياً مثل إيجاد الوسط الحسابي ، حساب معاملات الارتباط ومقاييس التشتت والترعة المركزية ... الخ من الخواص الإحصائية البسيطة والمعقدة .

ولهذا فقد دخل الحاسوب بإمكانياته الهائلة إلى هذا المجال العلمي الهام وأصبح استخدامه في العلوم الإحصائية ضرورة حتمية ، فأصبح يقوم بتهيئة البيانات وتصنيفها وعرضها بيانياً واستخدامها في وضع النماذج الرياضية الإحصائية وفي عمليات التنبؤ الاقتصادي والديموغرافي وإيجاد معادلات الانحدار والارتباط وجميع المعاملات الإحصائية ومتوسطاتها ، وكذلك وضع الدراسات الإحصائية في التحليل المالي والتحليل الإحصائي للعينات والاختبارات الإحصائية والرياضية المختلفة ...

وهناك العديد من البرامج الإحصائية الخاصة التي تقوم ببعض أو كل ما ذكر سابقاً إلا أن أفضلها هي (SPSS) و (SAS) ، إلا أن برنامج الإحصاء المحوسب (SPSS) هو البرنامج الأكثر شيوعاً والأفضل استخداماً في العلوم الإحصائية ، والذي سيدرس بالتفصيل في مادة الإحصاء المحوسب ، للتعرف على كيفية إدخال البيانات وحفظها وتحليلها واستخلاص النتائج . والشكل (14) يبين نافذة البرنامج الإحصائي SPSS .



الشكل (14) نافذة البرنامج الإحصائي SPSS

الفصل السادس

الشبكات الحاسوبية وتراسل المعلومات

Computer Network and Data Communication

(6 1) أهمية تراسل المعلومات:

استخدم الإنسان القديم، ومنذ العصور الأولى، عقله وذاكرته في التعلم واكتساب الخبرات، ومع مرور الزمن اكتشف أن واجبه وأهدافه لا تقف عند مرحلة التعلم وكسب المعرفة والخبرة، بل تعدى ذلك الى ضرورة نقل هذه الخبرات والمعارف الى الأجيال القادمة، عن طريق حفظ المعلومات على وسائل تخزين خارجية، وذلك لأن عقل الإنسان وذاكرته تنتهي بانتهاء حياته وقد تتأثر وتصاب بالخلل عند مرضه أو تعرضه الى أي ضغوطات أو تأثيرات نفسية أو أن تكون قدرته على حفظ المعلومات محدودة.

لقد تمثلت الذاكرة الخارجية قديماً لحفظ المعلومات وتراسلها باستخدام الواح الطين ثم الأحجار والأخشاب مروراً بالجلد وغيرها من الوسائل البدائية الى ما نشاهده اليوم.

ومع تطور الإنسان ودخوله عصر النهضة الصناعية الحديثة، انتشرت الوثائق التي تحمل المعلومات في شتى المجالات وتنوعت الوسائط، وتفجر بركان هائل من البيانات والمعلومات التي ليست بحاجة الى تخزين فقط بل الى سهولة التعامل معها ومعالجتها وسهولة استرجاعها باسرع وقت ممكن واقل تكلفة، فأصبح الإنسان يستخدم لتسجيل وتراسل المعلومات الورق على شكل كتب ومجلات وخرائط، ثم

ابتكر أسلوب البريد لنقل معلومات سياسية أو حرية أو اجتماعية أو انذارات، ومع مرور الزمن وتشعب الاعمال الحكومية والتجارية وتزايد حجم المعلومات صار التحدي الذي يواجه العلماء هو تحقيق الاتصال في الوقت المناسب، فتطورت اساليب نقل المعلومة وظهرت أنظمة الاتصالات المتمثلة بالهاتف والبرق والفاكس والتلكس وأخيراً الحاسوب بإمكانياته الهائلة ووسائله التخزينية المختلفة.

(6 - 2) الاتصالات والمعلوماتية:

منذ فجر الأربعينيات من القرن العشرين، بينما كان الحاسوب يظـهر الى الوجود، كانت أنظمة الاتصالات راسخة، وصناعة الاتصالات منتشرة في كل أنحاء المعمورة، ثم ما لبثت تكنولوجيا الحاسوب أن تطورت بسرعة هائلة من حيث تخزين البيانات واسترجاعها ومعالجتها فاصبح الحاسوب والمعلوماتية يشكلان قوة يصعب فصلهما عن بعضهما، وزاد من ترسيخ هذه القوة وأعطاهما بعداً فريداً من نوعه هو تطور الاتصالات وأجهزتها ووسائلها. وظهور ما يسمى بالشبكات. فما هو مفهوم الاتصالات الحديث..؟ وما هو مفهوم اتصالات البيانات..؟ وما هو مفهوم شبكة الاتصالات..؟ وعلاقة كل تلك المفاهيم بشبكة الانترنت..؟ هذا ما سنحاول ان نلقي الضوء عليه في هذا الفصل.

الاتصالات Telecommunications :

هي ارسال المعلومات بأي شكل مثل(صوت، بيانات، نصوص، صور، اشكال) من مكان الى آخر باستخدام الوسائل الإلكترونية أو الضوئية.

اتصالات البيانات Data Communications :

هو مصطلح أكثر تخصصاً، يصف عملية نقل واستلام البيانات من خلال خطوط الاتصال التي تربط بين حاسوب واحد أو أكثر ومعدات ادخال واخراج مختلفة.

شبكة الاتصالات Telecommunications Network:

هي أي ترتيب يقوم فيه المرسل بإرسال رسالة إلى مستلم عبر قنوات اتصال من نوع ما، والشبكة تحتوي على حواسيب مرتبطة ببعضها البعض وبعض وحدات طرفية تقوم بمعالجة وتبادل البيانات والمعلومات.

(6 3) الشبكة: (تعريفها أنواعها أهميتها)

الشبكة، من وجهة نظر المعلوماتية، هي مجموعة مواد ومعدات معلوماتية متصلة مع بعضها البعض، بكلام آخر: الشبكة هي عبارة عن مجموعة من الحواسيب Computers أو حاسوب كبير رئيسي ترتبط به مجموعة من طرفيات Terminals أو حواسيب صغيرة، ترتبط مع بعضها البعض بخطوط الاتصال (السلكية أو اللاسلكية)، بحيث يمكن استخدامها للمشاركة في الموارد المتاحة (من معلومات وبرامج) ونقل وتبادل المعلومات فيما بينهم.

فإذا كان الربط والاتصال بين الحواسيب بوسائط سلكية وفي منطقة محددة فعندئذ تسمى شبكة محلية LAN (Local Area Network)، أما إذا كان الربط عن طريق خطوط الهاتف أي باستخدام نفس مسار المكالمات الهاتفية في تبادل المعلومات الحاسوبية تسمى بالشبكة الواسعة WAN (Wide Area Network) وغالبا ما تكون ممتدة هذه الشبكة على مساحة جغرافية كبيرة وتستخدم عادة بين البنوك ومراكز البيع الواسعة وفي نظام الانترنت، وسنتناول في الفقرة التالية مكونات أنظمة الشبكات ومعداتنا المختلفة بالتفصيل.

إن أهمية الشبكات تأتي في أنها ساهمت وتساهم في رفع كفاءة التشغيل ودعم اتخاذ القرار في شتى مجالات الحياة من خلال:

1. كفاءة وسرعة الاتصال وسهولة نقل وتبادل المعلومات.
2. التشغيل الاقتصادي للأجهزة وذلك بالمشاركة في استخدامها.
3. المشاركة في البرمجيات.

4. المشاركة بالمعلومات المتوفرة وقواعد البيانات.
5. تطبيق مبدأ المعالجة الموزعة (Distributed Processing) التي تعني توزيع المهام على عناصر الشبكة مما يؤدي الى سرعة انجازها ورفع اقتصاديات تشغيل هذه العناصر.

(6 4) مكونات أنظمة الشبكات:

يمكن تقسيم العناصر المكونة لأنظمة الشبكات الحاسوبية من:

1. معدات الشبكات Network Hardware
 2. برمجيات الشبكات Network Software
 3. قنوات الاتصال Communication Channels
- وستتحدث عن كل من هذه المكونات بالتفصيل.

(6 4 1) معدات الشبكات: Network Hardware:

وتشتمل معدات الشبكات على الطرفيات ووحدات التحكم الرقمية ووحدات المواجهة البينية وأجهزة الحواسيب المختلفة:

أولاً: الطرفيات Terminals:

إن أي وحدة ادخال وإخراج مثل الطابعات-الراسمات-الأجهزة الصوتية-قارئ البطاقات المثقبة ، تدعى طرفيات، وذلك لوجودها في نهاية خطوط الشبكة، بالإضافة الى ذلك فإنه يطلق مصطلح الطرفيات على الحواسيب الصغيرة التي تكون في نهاية خطوط الشبكة وذلك لان الحاسوب يمكنه إرسال البيانات واستقبالها من حاسوب آخر شبيه به أو اكبر منه.

ثانياً: وحدات التحكم الرقمية Modems:

وهي عبارة عن جهاز يحول البيانات والمعلومات، من معلومات تناظرية (Analog) تسلك مسار المكالمات الهاتفية، الى معلومات رقمية (Digital) لكي تناسب الحاسوب لأنه كما نعلم ان الحاسوب والطرفيات تستخدم اشارات

عددية (Digital Signals) ذات جهود منفصلة تمثل بواسطة 1,0 بينما خطوط الهاتف تستخدم اشارات تناظرية (Analog Signals) ذات جهود متصلة، ولذلك قبل نقل الاشارات العددية عبر خطوط الهاتف يجب تحويلها الى اشارات تناظرية وتسمى هذه العملية بتعديل الاشارة Modulation وفي الطرف الآخر، عند استقبال البيانات يجب تحويل الاشارات التناظرية الى اشارات عددية يستطيع الحاسوب فهمها ومعالجتها، وتدعى هذه العملية بإعادة التعديل Demodulation ولذلك فإن وحدة التحكم الرقمية تستخدم للتعديل وفك التعديل.

ثالثا: وحدات الواجهة البينية Interface Units:

وظيفة وحدات الواجهة البينية توجيه عمليات ارسال واستقبال البيانات مما تزيد من كفاءة النظام بشكل كبير، ومن هذه الوحدات:-

1. جهاز مضاعفة الإرسال: Multiplexer: عبارة عن جهاز يقوم بدمج الاشارة القادمة من عدة طرفيات في قناة واحدة بشكل متزامن، وبالتالي يمكن أن تشارك عدة طرفيات بنفس قناة الاتصال، اذ يقوم هذا الجهاز بتقسيم خط الاتصال ذو السرعة العالية الى مجموعة من القنوات ذات سرعة أقل.

عادة تتم مضاعفة الارسال من خلال اسلوبين هما:

أ- أسلوب المضاعفة الموجية: حيث يتم تقسيم قناة الاتصال الى عدة قنوات من خلال تقسيم الحزمة الموجية الى حزم موجية أصغر تستخدم كل منها كقناة اتصال مستقلة.

ب- أسلوب المضاعفة الزمنية: حيث يتم تخصيص قناة عامة لخدمة عدة قنوات بحيث يكون لكل من هذه القنوات وقت محدد يستخدم خلالها القناة العامة لتمرير بياناتها.

2. المجمعات أو المراكز Concentrators: وهي أجهزة تحتوي على معالجات متوسطة (Micro Processors)، ويمكن أن تكون جهاز حاسوب صغير أو متوسط، وظيفتها دمج وتجميع البيانات الواردة إليها من عدة طرفيات وتخزينها بشكل مؤقت ثم إرسالها إلى جهتها المطلوبة عبر خطوط اتصالات سريعة.

3. مفاتيح الرسائل: Message Switchers: وهو عبارة عن جهاز يستقبل جميع الرسائل من الطرفيات المختلفة، ويقوم بتحليل الرسائل ومعرفة هدفها ومسارها الصحيح ثم توجيهها إلى الهدف أو الموقع المطلوب. نذكر هنا أنه هناك مفاتيح رسائل تمتلك القدرة على تخزين الرسائل إذا لم يكن خط النقل المطلوب متوفراً لحظة وصول الرسائل مما يخفف ذلك على الحاسوب الرئيسي مهمة توصيل الرسائل ويزيد من كفاءة الشبكة.

رابعاً: الحواسيب Computers

تختلف الحواسيب في الشبكة من حيث الحجم والقدرات والإمكانيات، وعادة في الشبكات الكبيرة يكون الحاسوب الذي يمثل مركز المعالجة الرئيسي من النوع الكبير إما في الشبكات الصغيرة فيكون من النوع الصغير.

ويدعى الحاسوب الذي يمثل مركز المعالجة الرئيسي بالحاسوب المضيف Host Computer أي الحاسوب الذي يستعمل الخدمات المقدمة من الحواسيب الصغيرة التي بدورها تدعى الحواسيب الملقمة.

(6 4 2) برمجيات الشبكات: Network Software

وتتكون برمجيات الشبكات من برمجيات التحكم في الاتصالات بين كل من الحاسوب المضيف والحواسيب الطرفية بالإضافة إلى برمجيات المراقبة وتوجيه البيانات والتحكم في تدفقها عبر خطوط الشبكة.

وتوفر حزم البرمجيات المستخدمة في الشبكات مجموعة من الوظائف الهامة منها:

1. رقابة فتح واغلاق خطوط النقل في الوقت المناسب.
2. تحديد أخطاء نقل المعلومات وتصميمها او اعادة نقلها عند حدوث خطأ.
3. تحديد مسار نقل البيانات وترتيب عمليات النقل بتسلسل معين حسب أولويات محددة.
4. حماية الشبكة من الاستخدام الخاطئ غير المسموح به.

(6-4-3) قنوات الاتصال Communication Channels

وهي المرات التي يتم فيها تراسل المعلومات والبيانات، وتسمى أيضا بـخطوط البيانات Data Links أو خطوط الاتصال Communication Links وترتبط هذه الخطوط مختلف الحواسيب والطرفيات والأجهزة الأخرى في الشبكة مع بعضها البعض. ولهذه القنوات أنواع مختلفة نصنفها ضمن مجموعات حسب صفات معينة وهي:

أولاً: حسب تركيبها الفيزيائي:

وتقسم إلى ثلاثة أنواع وهي:

1. الكوابل المعدنية Wire Cables

وهي التي تستخدم في الخطوط الهاتفية، هذه الكوابل تحتوي على حزم من اسلاك النحاس المجدولة (Twisted Pairs)، تحمل المعلومات على شكل إشارات تناظرية، وتتميز بكلفتها المنخفضة و توفرها في جميع انحاء العالم وسهولة استخدامها ولكن يعاب عليها بحساسيتها للتشويش الإلكترومغناطيسي مما يؤدي الى الحد من سرعة نقل البيانات عبرها. لذلك تم اختراع كوابل جديدة للتغلب على هذه المشكلة تدعى الكوابل المحورية Coaxial Cables وتتكون من كابل مركزي يحيط به حزمة اسلاك خارجية كدرع واق تحمي الكابل المركزي وتقلل من التشويش الإلكترومغناطيسي، وتمتاز أيضا بسرعتها في نقل المعلومات وامكانية استخدامها تحت الارض وتحت الماء.

2. الامواج المصغرة Micro Waves

وهي نفس أمواج الراديو بتردد عال، وتستخدم لنقل البيانات مع استخدام محطات تقوية تلتقط هذه الموجات ثم تعيد بثها بعد تقويتها، مما يسمح بنقلها الى مسافات بعيدة، ويمكن استخدام هذه الموجات أيضا في اتصالات الأقمار الصناعية Communications Satellite التي تلتقط الأمواج المصغرة من احدى محطات الأرض وترسلها الى محطات أخرى.

3. كوابل الألياف الضوئية Fiber Optic Cables

وهذا النوع يتكون من حزمة من الموصلات الزجاجية المصنوعة من السيلكون النقي والقادرة على نقل الضوء على شكل نبضات فعالة وغير فعالة (On and Off pulses) تولدها أشعة الليزر.

ثانيا: حسب كمية البيانات التي يمكن إرسالها عبر هذه القناة في وقت معين.

وهذا يعتمد على عرض حزام القناة (مدى الترددات التي يمكن نقلها)، ويمكن تصنيف القنوات حسب عرض الحزام الى:

الحزام الضيق Narrow Band : أقل من 2000 هرتز ونسبة النقل فيها تتراوح من 7 الى 30 رمز / ثانية.

الحزام الصوتي Voice Band : 3000 هرتز ونسبة النقل فيها حوالي 960 رمز / ثانية.

الحزام العريض Board Band : أكثر من 3000 هرتز ونسبة النقل فيها تصل إلى 40 مليون رمز / ثانية، وكوابل الامواج الألياف الضوئية والأمواج المصغرة هي قنوات الحزام العريض.

ثالثا: حسب اتجاه نقل البيانات: وتنقسم الى:

القنوات البسيطة Simplex Channels: والتي يتم فيها نقل البيانات باتجاه واحد فقط ولنقل البيانات باتجاهين يجب استخدام قناتين مستقلتين من هذا النوع واحدة للارسال والاخرى للاستقبال.

القنوات نصف المزدوج Half-Duplex Channels: وتقوم أيضا بنقل البيانات باتجاه واحد فقط ولكن يمكن تغيير اتجاه عملية النقل، أي أن النقل يتم أولا باتجاه واحد ثم يتم عكس الاتجاه ونقل البيانات في الاتجاه المعاكس.

القنوات الكامل الازدواجية Full-duplex Channels: تتم عملية النقل في كلا الاتجاهين في الوقت نفسه، مثل خطوط الهاتف.

رابعا: حسب أسلوب نقل البيانات: وهناك أسلوبان:

النقل المتزامن Asynchronous: وهذا يتطلب ارسال رمز واحد في الوقت الواحد، ولفصل الرموز عن بعضها يضاف بت البداية (Start bit) و بت النهاية (End bit) الى طرفي شيفرة الرمز المكونة من (8bit).
النقل المتزامن synchronous: ويتضمن نقل كتلة من الرموز عشوائيا، وهذه الكتلة من الرموز ترسل بدون بت البداية والنهاية، ويتم توقيت عملية النقل بحذر شديد.

خامسا: حسب طريقة ربط القنوات التي تربط الطرفيات بالحاسوب:

قنوات الاتصال المباشر Point-To-Point Channels: أي تربط كل طرفية مع الحاسوب عبر قناة منفصلة، وهذه الطريقة غالية الثمن، وتكون هذه الطريقة مجدية اقتصاديا عندما تكون كمية

البيانات المنقولة كبيراً جداً أو في الحالات التي تتطلب استجابة سريعة.

قنوات الاتصال المتعدد Multidrop Channels: تربط عدة طرفيات مع الحاسوب عبر قناة واحدة، وهذه الطريقة اقتصادية إذ تشترك عدة طرفيات بقناة واحدة ووحدة واجهة بينية واحدة، نشير هنا الى أنه في هذا الأسلوب من الربط يمكن لعدة طرفيات استلام البيانات بالشكل المتزامن أما لارسال البيانات فلا يستطيع الا طرفية واحدة فقط.

(6 5) شبكات المعالجة الموزعة

Distributed Data Processing Networks

بشكل عام يوجد نوعان من الشبكات هما:

1. شبكات الاتصالات Communication Networks: وظيفتها نقل البيانات وهذه تستخدم الحواسيب للمساعدة في عمليات النقل.
2. شبكات المعالجة الموزعة Distributed Processing Networks: وهي الشبكات التي تربط بين مكونات مادية مختلفة بهدف المشاركة في الموارد وامكانيات المعالجة، أي يكون لدينا عدة حواسيب كل منها يقوم بعمليات المعالجة الخاصة به بدلاً من حاسوب واحد يتحمل جميع اعباء المعالجة، ويمكن أن تحتوي الشبكة على حاسوب مركزي يقوم بعمليات المعالجة المشتركة مما يؤدي الى تفرغ كل حاسوب لانمام عملية المعالجة الخاصة به.

(6 6) هيكلية الشبكات Networks Topology

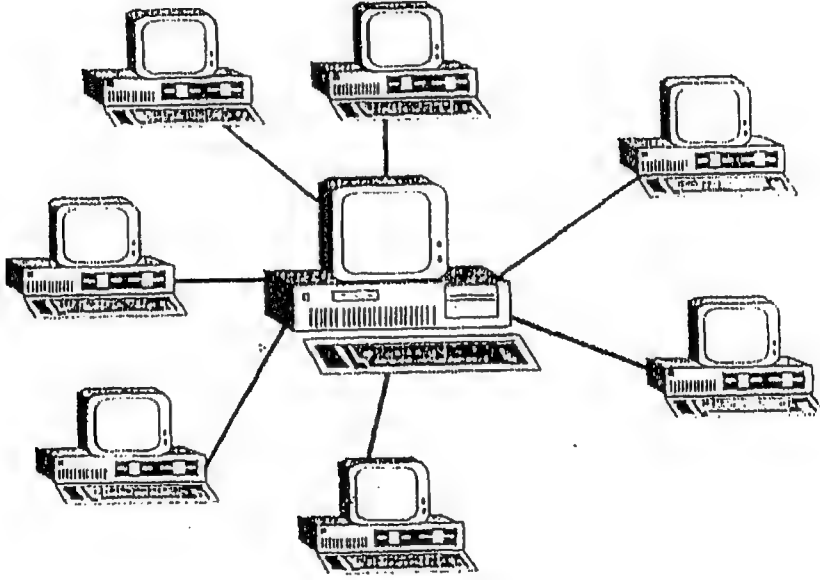
المقصود بهيكلية الشبكات هو نوع وشكل الاتصال بين المكونات المادية للشبكة ومن اهم واشهر هذه الاشكال:

1. الاتصال النجمي Star Topology (الشبكة النجمية)

2. الاتصال الحلقي Ring Topology (الشبكة الحلقية)
3. الاتصال الخطي Bus Topology (الشبكة الخطية)
4. الاتصال الهرمي Hierarchical Topology (الشبكة الهرمية)

(6-6-1) الشبكة النجمة Star Network:

وهي ابسط وأقدم أشكال الشبكات، وهي تتكون من حاسوب مركزي يسمى الخادم Server يتصل به مجموعة من الحواسيب الشخصية أو الطرفيات كما في الشكل (1)، حيث يعمل الحاسوب Server كنظام تحكم يتم من خلاله السيطرة على كافة الاتصالات بين الاجهزة المتصلة به، وأي انتقال للمعلومات يتم عن طريقه.



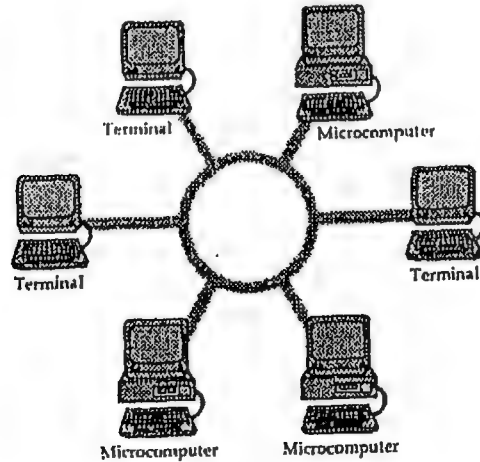
الشكل (1) نموذج الشبكة النجمية

ومن عيوب هذه الشبكة انخفاض درجة الموثوقية بها بسبب أن تعطل الجهاز المركزي يؤدي الى تعطل الشبكة بأكملها، اضافة الى ذلك أنه هناك ضياع للوقت بسبب زيادة وقت الانتظار بالنسبة للحواسيب الطرفية وذلك بسبب عدم امكانية انجاز اكثر من اتصال واحد في الوقت نفسه.

(2-6-6) شبكة الحلقة Ring Network

وهذه الشبكة تتكون من عدة حواسيب كل منها يتصل بالآخر مباشرة، وتتصف هذه الشبكة بعدم الحاجة لوجود حاسوب مركزي، لذلك فالحواسيب تأخذ شكل دائرة او حلقة كما في الشكل(2)، وهذه الشبكات تستخدم في المنشآت التي لا تحتاج الى تحكم مركزي لفروعها الموزعة، ويتم تنسيق الاتصالات خلال هذه الشبكة باستخدام بروتوكول خاص، والبروتوكول هو عبارة عن (مجموعة من القواعد والقوانين والاجراءات تحدد كيفية عمل الشبكة ووظائفها)، يسمى بروتوكول علامة المرور Token-Passing Protocol، هذه العلامة هي عبارة عن اشارة تحكم يتم بواسطتها تحديد الجهاز المسموح له بنقل المعلومات عبر الشبكة، وبالتالي حسب هذا المبدأ فان جهاز واحد فقط يستطيع استخدام الشبكة في الوقت نفسه.

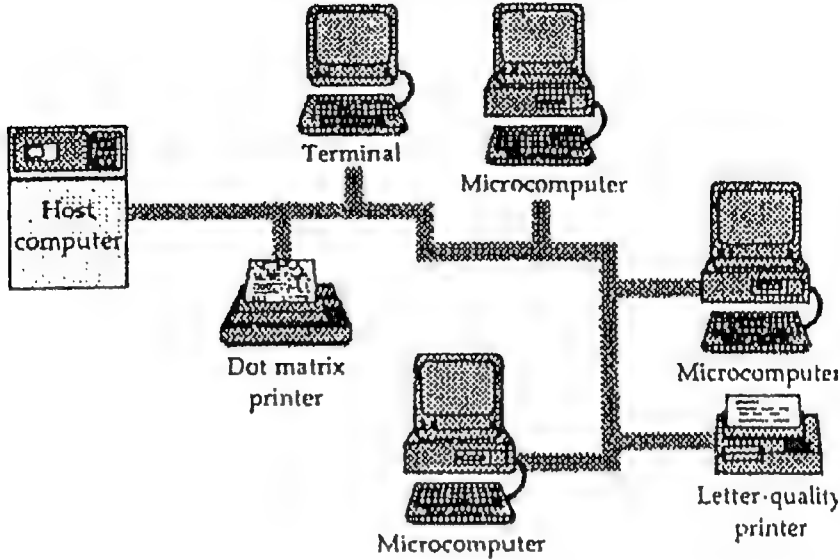
لذلك هذه الشبكة بدرجة عالية من الموثوقية، وذلك لأن تعطل أحد الأجهزة في الحلقة لا يؤثر على عمل باقي الاجهزة، بالاضافة الى قابليتها للتوسع واطرافه مكونات اخرى. والشكل (2) يمثل نموذجاً للشبكة الحلقية.



الشكل (2) نموذج لشبكة الحلقة

(3-6-6) الشبكة الخطية Bus Network

في هذا النوع من الشبكات ترتبط الاجهزة ومحطات التشغيل والطرفيات بواسطة كابل اتصال خطي Bus مفتوح ومزدوج الاتجاه، كما في الشكل (3).



الشكل (3) نموذج للشبكة الخطية

وللتحكم في مرور المعلومات خلال هذه الشبكة، فانه يستخدم بروتوكولات خاصة تسمى Carrier-Served Access اكتشاف التصادم و اختصارا (CASM) ومن هذه البروتوكولات:

- بروتوكول مانع التصادم (CSMA/CA) (Collision Avoidance) والذي يتضمن قيام جهاز واحد بنقل المعلومات عبر الشبكة في الوقت نفسه.
- بروتوكول اكتشاف التصادم (CSMA/CA) (Collision Detection)، لتنظيم عملية النقل حسب الاسبقية في طلب خدمة النقل لمنع التشويش.

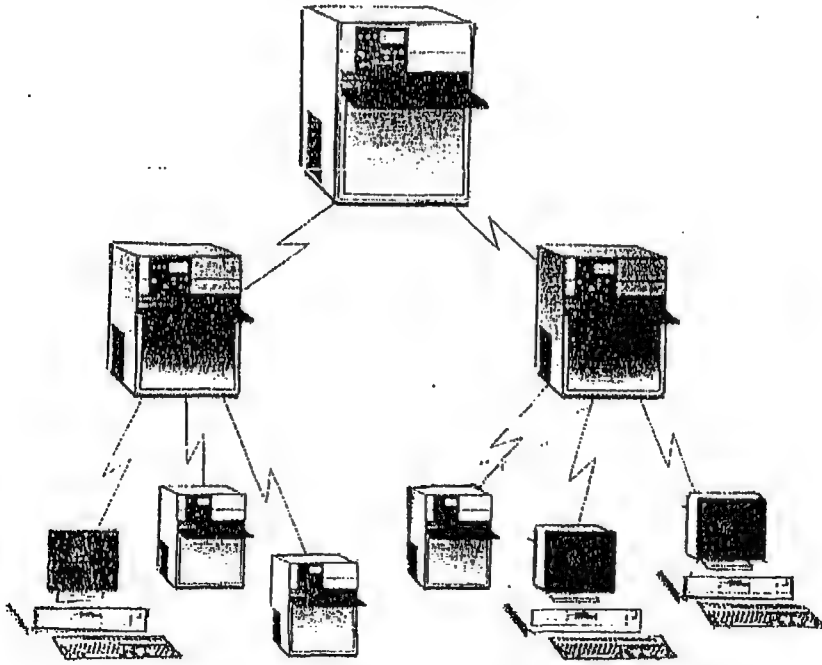
مميزات هذه الشبكة:

1. البساطة مع المرونة والقابلية للتوسع.

2. توفير البروتوكولات المنظمة لعمل الشبكة.
3. درجة الموثوقية عالية فيها حيث أن الشبكة لا تتوقف عن العمل الا بتوقف إحدى محطاتها.

(4-6-6) الشبكة الهرمية Hierarchical Network

تتصل الحواسيب فيها على شكل هرم حسب مستوى أهميتها، والشكل (4) يسمى الحاسوب في قمة الهرم بالحاسوب الجذر (Computer Root) وهو يتحكم في الحواسيب التابعة له، وهي بدورها تتحكم بالحواسيب المتفرعة منها وهكذا، ولذلك تدعى أحيانا باسم الشبكات السيد/الخادم (Master/Slave Network) وعادة يستخدم هذا الأسلوب في المنشآت التي تعتمد إدارتها على سلسلة من الأوامر والتعليمات وكما في الشكل (4).



الشكل (4) نموذج لشبكة هرمية

(6-7) الانترنت Internet

(6-7-1) ماهية الانترنت

الانترنت هي شبكة معلوماتية عالمية مؤلفة من مجموعة كبيرة من شبكات الحاسوب موصولة مع بعضها البعض، لقد كانت البدايات الأولى لهذه الشبكة مقتصورة على عدة اجهزة حواسيب، لا يزيد عن عشرة حواسيب مضيفة، ثم ما لبثت ان تطورت وتوسعت واصبح بإمكان أي شخص من خلال حاسوبه الشخصي في المكتب أو البيت أو في أي مكان قابل للربط مع هذه الشبكة الحصول على المعلومات التي يحتاجها من مصادرها وحتى وان كانت هذه المصادرة في أقصى بقاع العالم شريطة أن تكون هذه المصادر مرتبطة بهذه الشبكة العالمية، والتي ترتبط بدورها بملايين الحواسيب ومراكز المعلومات المنتشرة في العالم لتشكل ما يعرف بالنسيج العالمي الواسع للمعلومات او شبكة المعلومات العالمية .World Wide Web (WWW).

لقد كانت التجربة الأولى عام 1969م، عندما تم توصيل حواسيب وزارة الدفاع الامريكية، وبدأ كل منهم يتحدث مع الآخر عبر هذه الشبكة الصغيرة، وبعد نجاح هذه التجربة تم توصيل حاسبات هذه الشبكة مع جهات اخرى، وحققت أيضا نجاحا باهرا وأخذ عدد الأجهزة الحاسوبية المتصلة مع هذه الشبكة تزداد، ثم نشأت مجموعة من الشبكات، واصبحت شبكة الانترنت تجمع مجموعة من شبكات المعلوماتية المختلفة، وهنا نشير الى ان شبكة الانترنت ليست هي الوحيدة في العالم بل هناك شبكات عديدة تهتم بالمعلوماتية منها شبكة كمبيوسيرف (CompuServe) وشبكة أمريكا أون لاين (America on Line) وشبكة بت نت (Bit Net) وهذه الشبكات تملكها شركات مقرها الولايات المتحدة الامريكية، مسؤولة عن تعديل المعلومات فيها بشكل مستمر والتحكم بنوعية خدماتها ومعلوماتها.

أما شبكة الانترنت فوضعها مختلف تماما، اذ ان هذه الشبكة لا تملكها أي جهة معينة في العالم، ولا سلطان عليها من أي شخص، ولا هيمنة أي منظمة دولية، بل انما ملك للجميع، يستطيع أي شخص الاتصال بها ويخصص له عنوان ويصبح جزءا منها وشريكا فيها.

يعود انتشار هذه الشبكة و توسعها لعدة أسباب أهمها:

- 1- الوصول المجاني الى المعلومات، الا في حالات قليلة، حسب أهمية المعلومات، وما تدره من أرباح، أما بخصوص الأموال التي تدفع الى الجهات الموزعة للخدمة عند الاشتراك في خدمة الانترنت فهي لصيانة الخطوط وزيادة سرعتها وبث البرامج الموصلة للخدمة.
- 2- الزيادة المستمرة في عدد المشتركين، وزيادة موارد الخدمات والدول المشتركة على الشبكة، مما يزيد من كميات المعلومات المتاحة.
- 3- بناء الشبكة على أساس نظام (العميل/الخادم) (Client/Server) هذا النظام في خدمات الانترنت يعطي إمكانية توصيل جميع الحواسيب الى الشبكة، بحيث يسمح لها بإمداد خدمات واستلام خدمات من بعضها البعض.

ملاحظة:

* العميل Client: هو الذي يتعامل مع المستخدم، ومن مسؤولياته قبول مدخلات لوحة المفاتيح واطهار البيانات على الشاشة للمستخدم.

* الخادم Server: مسؤول عن انجاز جميع العمليات الحاسوبية بواسطة العميل مثل استرجاع البيانات بالنيابة عن العميل وانجاز العمليات الحاسوبية، وإرسال البيانات الى العميل.

* نظام (العميل/الخادم) يمكن تشغيله على حاسوب واحد، الا انه في الغالب يتم تشغيله على حاسوبين مختلفين، احدهما عميل والاخر خادم.

(6-7-2) لحة تاريخية عن شبكة الانترنت:

سنعرض في هذه الفقرة تطور شبكة الانترنت منذ بدايتها وحتى الآن،

وأهم شبكات المعلومات المتصلة معها وما تقدمه من خدمات:

1. ما قبل تاريخ الشبكة: حيث أنشأت وزارة الدفاع الأمريكية

DOD (Department Of Defense) عام 1957 وكالة لمشاريع الأبحاث

المتقدمة Advanced Research Project Agency (ARPA) فتم

بتطوير العلوم التي تخدم الاحتياجات العسكرية. ويشار الى ان تاريخ الانترنت

بدا في الولايات المتحدة في الستينات وكانت عبارة عن شبكة تعتمد طريقة

تبديل الحزم وهي تقنية نقل معلومات تركز على تقسيم الرسالة الى سلسلة من

الحزم التي ترسل عبر الشبكة.

2. أربانت **ARPA net**: جرى أول تحقيق عملي لشبكة أربانت في جامعة

كاليفورنيا في لوس انجلوس 1969 و تألفت الشبكة من أربعة حواسيب، ثم

أضيفت عقد أخرى الى الشبكة وبشكل تدريجي وخاصة عقد من معهد بحوث

ستانفورد في جامعة كاليفورنيا في سانتار باربارا وكذلك عقد من جامعة يوتا،

حتى اصبح عدد العقد 62 موقعا عام 1974 و عام 1981 اكثر من 200

موقع.

إنترنت **Internet**: جرى عرض أول تحقيق تجريبي عملي يتضمن 40 عقدة

لشبكة أربانت خلال المؤتمر العالمي الأول حول الاتصالات الحاسوبية في واشنطن

عام 1972 وتركزت المناقشات حول توحيد بروتوكولات الاتصالات، و نتيجة

لهذا المؤتمر تم وضع المخطط الأول لبنية الشبكات العالمية المتحدة في المستقبل حيث

تمتتع الشبكات الجزئية باستقلالية كبيرة وتتصل فيما بينها. وفي عام 1979 تم

انشاء مجلس للتشكيل والتحكم بالانترنت هدفه مراقبة تطور الشبكة ويسمى:

Internet Configuration Control Board (ICCB)

4. ميلنت **MIL net**: قسمت شبكة اربانت عام 1983 الى شبكتين اربانت وميلنت، ارتبطت شبكة ميلنت بشبكة المعطيات الدفاعية، أما اربانت فقد كانت العمود الفقري لشبكة إنترنت عام 1990 ومن ثم ضمت اربانت الى شبكة القاعدة العلمية الوطنية (NSF net) National Science Foundation Net التي أوضحت بدورها العمود الفقري لشبكة إنترنت بين عامي 1990 1995، وفي عام 1995 بدلت هذه الشبكة بمجموعة شبكات كبيرة متصلة مثل , ANS net Sprint net , MCI net

5. يوزنت **USE net**: هي ليست شبكة بل خدمة مؤتمرات الكترونية تستخدم الشبكة كحامل للمعلومات. نشأت يوزنت عام 1979 وكانت على شكل ثلاث حواسيب هدفها نقل الاخبار بين مستخدمي نظام يونيكس، واستخدم بروتوكول UUCP (Unix To Unix Copy): وهو بروتوكول لتبادل الرسائل والملفات إلكترونيا بين مستخدمي نظام Unix، سرعان ما ازداد عدد مواقع اربانت المستثمرة لأخبار يوزنت الأمر الذي أدى الى استبدال بروتوكول UUCP بروتوكول نقل اخبار الشبكة

(NNTP) : (Net News Transfer Protocol).

6. بت-نت **Bit net**: أنشأت في جامعة نيويورك لتأمين نظاما للمؤتمرات الالكترونية ولكن سرعان ما دجت مع شبكة CS net عام 1989 لتشكيل شبكة التعاون من أجل شبكة البحوث والتعليم

Corporation for Research and Education Network (CREN)

7. شبكات الخدمة الآنية: تكون هذه الشبكات متصلة مع شبكة إنترنت وتقدم خدمات آنية لمستخدميها على المستوى القومي والعالمي وتوجد بين هذه الشبكات وشبكة الانترنت عبارات للرسائل، ومنذ فترة وجيزة أمنت معظم هذه الشبكات لمستخدميها ولوجا الى شبكة الانترنت الى حد ما. أهم هذه الشبكات:

أ. كمبوسرف **CompuServe** : تأسست عام 1979 تؤمن هذه الشبكة خدمات البريد الالكتروني والمناقشات على الشبكة ونقل الملفات ويتم الوصول مع الشبكة بواسطة مودم.

ب. امريكا أونلاين (**AOL America Online**) :

تعتبر إحدى أكبر الشبكات الأمريكية وخاصة بعد شرائها لعدة شركات أخرى مثل: **ANS net** في شباط 1995، التي تعتبر إحدى شبكات إنترنت الأساسية في أمريكا، وتهتم شركة **AOL** بالأمور الإعلامية.

ج. دلفي إنترنت **Delphi Internet**: تهتم بالمعلومات الإعلامية منذ عام 1995 اذ تتضمن عددا كبيرا من الصحف العالمية الهامة مثل (**Sun, New**) (ومحطات تلفزيونية (**Fox** الشبكة الأمريكية الرابعة، **BSKYB** في انكلترا) وأقمار صناعية وشركات سينمائية

د. بروديجاي **Prodigy**: تعتبر هذه الشبكة فرع مستقل من **IBM** و **Sears**، أنشأت عام 1988.

هـ **Internet MCI**: التي وفرت منذ زمن طويل خدمة البريد الالكتروني (**MCI Mail**).

و. شبكة مايكروسوفت **Microsoft Network**: تؤمن خدمة البريد الالكتروني وخدمة نقل الملفات وخدمة المناقشات وعدد كبير من الخدمات التجارية لكل حاسوب يعمل تحت ويندوز 95 ومرتبطة مع مودم أو مع ويندوز 98.

ز. **Apple Work**: شبكة خدمة آنية لشركة **Apple**.

ح. ويل **Well**: تأسست هذه الشركة لتأمينولوج الى الانترنت.

(6-7-3) خدمات الانترنت

خدمات الانترنت كثيرة جدا ولذا سوف نقتصر على أهم هذه الخدمات:

1. خدمات الاتصالات:

أ. البريد الالكتروني (Electronic Mail) وفي هذه الخدمة يمكن استقبال وارسال الرسائل من و الى أي شخص في العالم يستخدم هذه الشبكة ان أهم البرامج المستخدمة في البريد الالكتروني هي:

- ايدوارا (Eudora)

- تيلنت (Telnet)

للدخول الى برنامج (Eudora) يتم بالشكل التالي:

(com /> . (Eudora , < http://www.qualcom

ب. مجموعات النقاش: تعود شبكات الاخبار (News) على الشركات بفائدة كبيرة، فهي تحوي على معلومات متعددة، كما أنها تسمح بتأمين وصل مباشر بين الشركات وزبائنهم، مما يؤدي الى تحسين الخدمات التي تقدمها الشركة، اما الافراد فيمكنهم استخدام الأخبار News للتسلية أو لإجراء مقابلات او اعلانات صغيرة

News (Fr.Petites annonces. Vehicules) أو استدراج

عروض News : Fr. Jobs .offers.

ج. خدمات حوار إنترنت : تسمح (IRC) Internet Relay Char

لمستخدمي إنترنت بالتخاطب في الزمن الحقيقي بشكل نصي، يمكن

الاستفادة منها في التطبيقات المهنية او الخاصة، اما Internet Phone

فتسمح باجراء محادثات هاتفية باستخدام شبكة إنترنت.

2. خدمات الدعاية والاعلان

أ. الخدمات المجانية: معظم المعلومات المتوفرة على الشبكة تقدمها الهيئات

التي تنتجها مجاناً. وتقدم معظم الشركات المعلومات خاصة بمنتهجها

وكمثال على ذلك فان شركة مايكروسوفت تقدم للجميع تصميمات برمجية Patches وبرمجيات مجانية ومعلومات فنية عن منتجاتها، ومعلومات حول الشركة ذاتها مثل (التقرير السنوي، اعلانات الصحافة، عروض العمل، ... الخ).

كذلك تنشر الجامعات والمدارس ومراكز البحوث كمية من المعلومات ان معظم المكتبات الضخمة تتواجد أيضا على الانترنت، كما تنشر وكالة الاستخبارات المركزية CIA كتابها عن الجرائم على الانترنت.

<<http://www.odci.gov/cia/publications/pubs.html>>

كما تضع وكالة الفضاء الـ NASA أخبار مهماتها السابقة والحالية واللاحقة وتنشر معلومات خاصة بالملكوك الفضائي.

<<http://shuttle.nasa.gov/>>

كما تعرض المجموعة الاوربية وثائق اقتصادية وسياسية

<<http://www.echo.lu/>>

ب. الخدمات التجارية: تقدم بعض المؤسسات او شركات الأنباء المحلية عبر شبكة الوب أو عبر News مثل الاخبار المالية وأسعار البورصة ونسب صرف أسعار العملات ويمكن ان تكون شبكة الانترنت وسيلة إعلام جديدة قادرة على مناقشة أو تكميل وسائل الاعلام التقليدية كالصحافة والمذيع أو التلفاز.

ج. النشر الآلي: هناك عدد من المجلات التي تنشر إلكترونيا مثل عالم إنترنت Internet World على العنوان:

<<http://www.mecklesweb.com:80/mags/iw/iwhome.htm>>

كما ان معظم الصحف الامريكية موجودة على العنوان:

<<http://www.pathfinder.com>>

د. بنوك المعطيات: يمكن الوصول بسهولة الى العديد من بنوك المعطيات عن طريق الانترنت.

3. الخدمات المساعدة: تصنف ضمن هذه الفئة كافة الخدمات عن بعد والتي تهدف الى بيع منتج ما او خدمة عن طريق الشبكة، حيث نجد اليوم على الشبكة كل شيء من ازهار مقصوصة الى بياضات، مروراً بالكتب والبرمجيات والسيارات... الخ.

4. الثقافة والسياحة: نجد على شبكة WWW متاحف تعرض قسم من المجموعات الأثرية و المعارض المؤقتة و معلومات عامة مختلفة (مثل التوقيت، مخطط السير ضمن المعرض التعرف... الخ

5. التعليم والبحث: يستعمل المدرسون و الباحثون الانترنت ليتبادلوا بسرعة كبيرة أعمالهم، مثل نتائج تجاربهم او مصادرهم المعلوماتية والتربوية. كما ان الانترنت هي مصدر هائل للمعلومات للمدرسين والطلاب، حيث يستطيعون الحصول على المعلومات المناسبة وهم وراء اجهزتهم.

6. الألعاب: يمكن لرائد الانترنت تحميل العاب عن بعد عبر البرامج: Freeware, Shareware.

(6-7-4) الاتصالات والعنونة على شبكة الانترنت:

بروتوكولات الاتصالات Communication Protocols:

ان الحديث بين شخص وآخر هو نوع من الاتصال، ولكي يتم الاتصال بطريقة صحيحة لا بد من مراعاة عدة قواعد، مثل ان تكون لغة التخاطب معروفة لكلا الطرفين، عندما يتكلم احد الطرفين لابد ان ينصت الآخر.. الخ هذا النوع من القواعد ينطبق أيضا على الاتصال بين الحاسبات، فعند اتصال جهازي حاسوب معا، لابد أن تحكم اتصالاتهما مجموعة من القواعد لكي يتم الاتصال

بنجاح وهذه المجموعة من القواعد تسمى بروتوكولات الاتصال
Communication protocols.

إن الاختلاف بين الحاسبات المتصلة بشبكة الانترنت وبين انظمة تشغيلها
وبرامجها، وبين الكابلات الموصولة عليها، أدى الى ايجاد بروتوكول للتحكم في
الاتصالات وهذا ما أطلق عليه البروتوكول القياسي لاتصال الانترنت
(Transmission Control Protocol/Internet Protocol)(TCP/IP)
كما يوجد هناك العديد من البروتوكولات نذكر منها:

HTTP: بروتوكول نقل النصوص(الصفحات الإعلامية) Hypertext
Transfer Protocol

FTP : بروتوكول نقل الملفات File Transfer Protocol

SMTP: بروتوكول البريد الإلكتروني Simple Mail Transfer Protocol

Internet Addressing العنونة على شبكة الانترنت

من المعروف انه للاتصال بأي شخص آخر تلفونيا لابد من معرفة رقمه،
وكذلك الحال بالنسبة للحاسبات، اذ لا يمكن الاتصال بأي حاسوب دون معرفة
عنوانه. لكل حاسب متصل مع شبكة الانترنت عنوان هذا العنوان يسمى Ip
Addresses وهو عبارة عن رقم مقسم الى أربعة أقسام A,B,C and B وكل
قسم يأخذ أحد الأرقام بين الرقم صفر والرقم 255 ولكن الرقم صفر والرقم 255
محجوزان دائما لأغراض خاصة بالشبكة مثال هذه الأرقام:

- 160.121.6.4
- 162.122.8.5
- 180.202.2.3

ونظرا لصعوبة حفظ هذه الأرقام تم استخدام الأسماء أو الأحرف بدلا منها وهذه
العناوين في حقيقتها ليست أسماء الا لدى المستخدم، اما بالنسبة للحاسوب فهي ما
زالت أرقاما على سبيل المثال.

العنوان بالنسبة للمستخدم Ritsec.com.eg

يعني بالنسبة للحاسوب 163.121.2.3

كما أن لهذه الاسماء أيضا قواعد لا بد من إتباعها وهي كالتالي:

1. لا بد أن يكون اسم الدولة الموجودة بها الحاسب وهو جزء من عنوان الحاسب، لذلك تم اختصار اسماء الدول الى حرفين مثل Egypt تأخذ Eg و United Arab Emirates تأخذ ae و France تأخذ Fr.

2. تحت اسم الدول يأتي اسم نطاق العمل وكل نطاق يتم اختصاره الى حرفين او ثلاثة أحرف على سبيل المثال:

- نطاق تجاري Commercial Sector اختصاره COM.

- نطاق حكومي Government Organization اختصاره Gov.

- نطاق تعليمي Educational Institutions اختصاره Edu.

- الاغراض العسكرية Military اختصاره Mil.

3. تحت اسم نطاق العمل يأتي أسم الشركة أو المكان التابع لهذا النطاق على سبيل المثال IBM اختصاره ibm.

4. تحت اسم الشركة يأتي اسم الحاسوب مثال ذلك:

Mustafa.cf.gov.ae

Daher@Cultwral.org.eg

لا بد من الإشارة هنا الى ان الرمز @ ينطق at (آت) و النقطة (.) تنطق Dot (دوت) و باقي الحروف كما هي.

(6-7-5) أسس الدخول الى شبكة الانترنت

من اجل الدخول على شبكة الانترنت لا بد من توفر العناصر الضرورية

التالية:

1. حاسوب شخصي طول الكلمة فيه 32 بت، مزود بجهاز مودم Modem ومزود أيضا بخط خاص مباشر (Leased Line).

2. أسم تعريفى وكلمة سر (اشترك في الانترنت).

3. برامج تتيح الدخول على خدمة الانترنت.

يمكن تقسيم طرق الدخول على الانترنت الى طريقتين:

1. طريقة غير مباشرة:

وهي الطريقة التي يسلكها معظم مستخدمي الانترنت إذ يكون الحاسوب الشخصي مزودا بفرع Terminal يربطه بالحاسوب الرئيسي يكون متصلا مباشرة بالانترنت.

2. طريقة مباشرة:

وهي أن يكون الحاسوب الشخصي عقدة مستقلة Individual Node على شبكة الانترنت، وبذلك القدرة على ادخال أي مشتركين آخرين على الشبكة عن طريقة.

(6-7-6) البرامج المستخدمة في البحث وتصفح المعلومات عبر الانترنت

Internet Browsers

من أهم البرامج المستخدمة لعرض و البحث عن المعلومات هي:

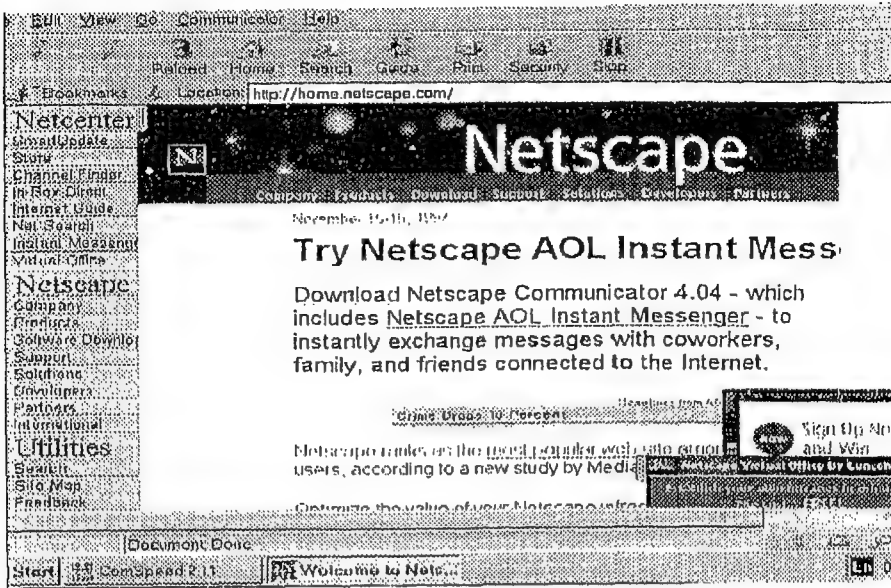
1. برنامج نتسكيب Netscape

2. برنامج موزايك Mosaic

3. برنامج اكسبلورير Explorer

برنامج نتسكيب Netscape:

هذا البرنامج يحتوي على تقنيات يمكن من خلالها الدخول الى شبكة الانترنت ويمكن من خلاله ارسال و استقبال الرسائل والأخبار والمعلومات، الشكل(5) يوضح الشاشة الرئيسية لهذا البرنامج.



الشكل (5) الشاشة الرئيسية لبرنامج Netscape

مجموعة مفاتيح برنامج Netscape:

- المفتاح Back: يرجع الصفحة الاعلامية السابقة.
- المفتاح Forward: يتقدم الى صفحة تالية.
- المفتاح Home: يظهر صفحة بداية تشغيل برنامج نتسكيب
- المفتاح Reload: خاص باعادة تحميل الصفحة مرة اخرى.
- المفتاح Image: خاص باعادة تحميل الصور داخل الصفحة اذا لم تكن واضحة عند تحميلها في المرة الاولى.
- المفتاح Open: خاص بفتح صفحة اعلامية اخرى.
- المفتاح print: يقوم بارسال الصفحة التي امامك الى الطابعة.
- المفتاح Find: خاص للبحث عن كلمة داخل الصفحة الموجودة فيها.
- المفتاح Stop: خاص لعدم اكمال تحميل الصفحة.
- المفتاح What's New: من خلال هذا الزر يستطيع المستخدم معرفة ما هي الاماكن الجديدة التي تم افتتاحها على شبكة الانترنت.

- المفتاح Whats Cool: لتثبيت الاماكن التي يكثر استخدامها على الشاشة.
 - المفتاح Destinations: يساعد في الحصول على اكثر الاماكن التجارية التي يستخدمها رجال الاعمال على شبكة الانترنت.
 - المفتاح Net search: يعرض اكثر من طريقة للبحث عن موضوع معين.
 - المفتاح People: خاص بالصفحات البيضاء White Pages، حيث بواسطته نستطيع البحث عن أي مستخدم داخل الشبكة بواسطة اسمه او عنوانه البريدي.
 - المفتاح Soft Ware: خاص ببرنامج Netscape وذلك من اجل الحصول على الاصدار الجديد للبرنامج او الاطلاع على منتجات شركة Netscape.
 - مستطيل العناوين Net site: يوضح هذا المستطيل عنوان الصفحة التي تظهر امامك على الشاشة، اذا اردت ان تنتقل الى صفحة اخرى او الى أي مكان اخر، تستطيع كتابة العنوان داخل هذا المستطيل ثم Enter.
- يمكن استخدام عدة انظمة من اجل البحث عن المعلومات في الانترنت باستخدام برنامج Netscape من اهمها:

أ. اكسايت Excite:

توفر هذا النظام شركة Architext Software، ويشمل أداتين للبحث، الاداة الاولى Net Search وهي تتيح المجال للبحث عن مواقع معينة في الشبكة او البحث في المجموعات الاخبارية Newsgroups لشبكة Usenet الاداة الثانية Net Reviews وهي توفر مخطط هرمي لمحتويات الشبكة مفهرسة حسب الموضوعات و تفرعاتها يمكن الوصول الى موقع Excite من خلال العنوان التالي:

<http://www.excite.com>

والشكل (6) يبين الشاشة الرئيسية لنظام Excite.



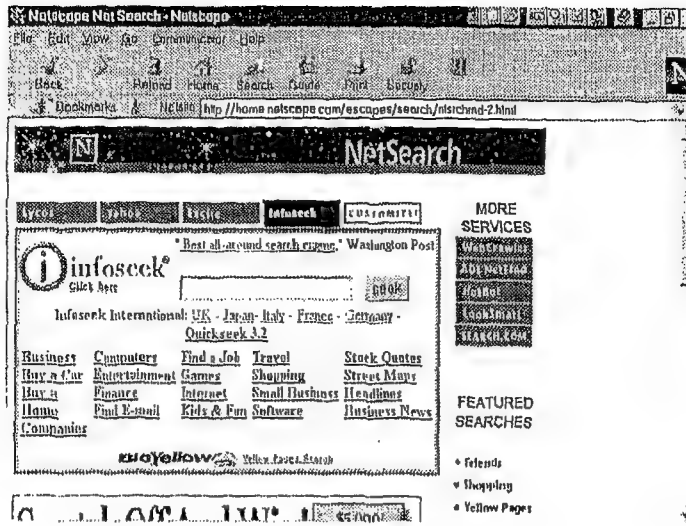
الشكل (6) الشاشة الرئيسية لنظام Excite.

ب. نظام انفوسيك Infoseek:

يوفر هذا النظام الخدمة على شكل مجاني و توفر هذا الشكل من الخدمة شبكة Usenet Web، وشكل مدفوع الآخر، يمكن الدخول على نظام انفوسيك باستخدام العنوان التالي:

<http://www.infoseek.com>

والشكل (7) يبين الشاشة الرئيسية لنظام انفوسيك ضمن برنامج Netscape



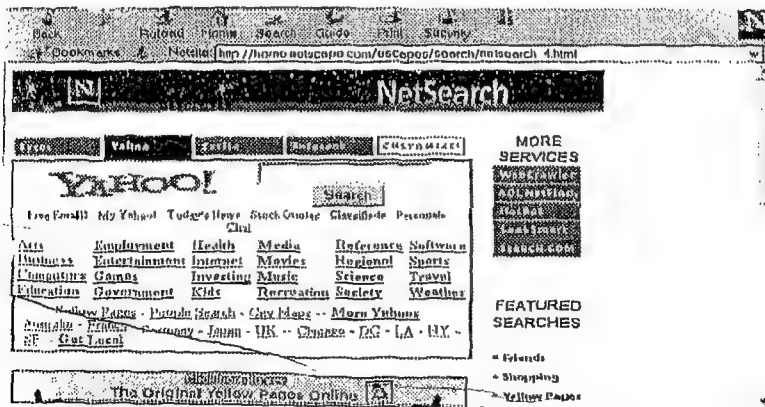
الشكل (7) الشاشة الرئيسية في نظام انفوسيك

جـ نظام خدمة ياهو Yahoo

وهو اكثر الأنظمة استخداما، حيث يحتوي على آليات بحث متطورة لأنه يحتوي على كلمات مفتاحية وفهرس منظم حسب المواضيع، ويحتوي أيضا على دليل يصف المواقع الأكثر شيوعا عند المستخدمين، يمكن الوصول الى نظام يساهو من خلال العنوان التالي:

<http://www.Yahoo.com>

والشكل (8) يبين الشاشة الرئيسية لنظام ياهو



الشكل (8) الشاشة الرئيسية في نظام ياهو

د. نظام خدمة ليكوس Lycos

يعد هذا النظام من أهم آليات البحث المتوفرة على الشبكة، حيث يتوفر فيه قاموس يشمل أهم الكلمات والمصطلحات في الشبكة تصل إلى 8 ملايين كلمة، ويضم أيضا آليات للبحث عن النصوص والصور والصوت، يمكن الوصول إلى موقع ليكوس من خلال العنوان التالي:

<http://www.lycos.com>

هـ نظام خدمة ماجلان Magellan

يحتوي هذا النظام قاموسا يحتوي على أهم العناوين على شبكات Gopher, FTP, Web, Usenet. وعنوان هذا النظام هو:

<http://www.mckinley.com>

و. نظام خدمة فهرس السياحة Web Servers Tourist Directory

يشمل هذا النظام تقنية متطورة لمتابعة المواقع العلمية والسياحية على خارطة العالم المعروضة على الشاشة، وعنوان هذا النظام:

<http://www.w3.org/hypertext/datasawrces/www/servers.html>

أسئلة وتمارين عامة

1. عرف الشبكة و ما هي أنواع الشبكات؟
2. ما هي أهم معدات الشبكات؟
3. ما هي أهم وحدات عمليات ارسال و استقبال البيانات؟
4. ما هي أهم قنوات الاتصال و ما هي أهم أنواع القنوات حسب تراكيبها الفيزيائي؟
5. اذكر أهم الشبكات المحلية للحاسوب؟
6. عرف شبكة الانترنت؟
7. اذكر أهم خدمات شبكة الانترنت؟
8. عرف بروتوكولات الاتصال واذكر أهمها؟
9. كيف تتم عملية العنونة على شبكة الانترنت؟
10. ماذا يلزمنا من اجل الدخول على شبكة الانترنت؟
11. تحدث عن الطريقة المباشرة و الغير مباشرة للدخول على شبكة الانترنت
12. ما هي أهم البرامج المستخدمة في البحث و تصفح المعلومات في الانترنت؟
13. ما هي أهم الأنظمة المستخدمة في برنامج نتسكيب؟

الفصل السابع

لغات البرمجة

Programming Languages

(7 - 1) المكونات والعناصر الأساسية للغة البرمجة :

اللغة البرمجية هي مجموعة من القواعد والاصطلاحات تستخدم لتمثيل المعلومات من قبل الانسان ويفهمها الحاسوب بعد ترجمتها بواسطة مترجمات او مفسرات الى لغة الآلة ، وتتكون لغة البرمجة مثل أي لغة انسانية من مجموعة من العناصر التالية : Beginner's All - Purpose Symblic Instruction code

(7-1-1) مجموعة الرموز الاساسية :

وهي مجموعة العناصر التي تكون اللغة ، وهي مكونة من كل أو جزء من الرموز الآتية :-

1- الحروف اللاتينية وهي :

A , B , C , D , E , ... X , Y , Z

2- الأرقام العربية وهي :

0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9

3- الرموز الخاصة وهي :

+ , - , * , / , \$, (,) , { , } , < , > , = , ? ,

(7-1-2) الكلمات

وهي نوعان :

أ- الكلمات المحجوزة : وهي كلمات لها معاني خاصة بالنسبة للحاسوب وعندما يتعرف الحاسوب عليها يقوم بتنفيذ الأوامر المناظرة لمعاني هذه الكلمات ومن

امثلتها : Read , Print , FOr , ...

لا يجوز للمبرمج إعادة تعريفها لأي غرض آخر في البرنامج .

ب- مسميات المبرمج التعريفية : وهي كلمات يختارها المبرمج في البرنامج لعنونة

البيانات مثل : sum, I,Z, qty,...

وتختلف قواعد كتابة مسميات المبرمج من لغة الى أخرى .

(7-1-3) مجموعة التعليمات والأوامر :

التعليمة : هي مجموعة من الرموز الخاصة بلغة البرمجة التي عندما يفهم الحاسوب معناها ومدلولاتها تستطيع وحدة التحكم فيه أن ترسل الاشارات التنظيمية لبقية وحدات الحاسوب القيام بما تمليه عليها مفهوم هذه التعليمات ، وتنقسم مجموعة التعليمات أو الأوامر التي تتضمنها أية لغة برمجية من الناحية الوظيفية الى الانواع التالية :

1. تعليمات وأوامر الإدخال : وهي التي تقوم بقراءة أو إدخال البيانات المدخلة وتخزينها في وحدة التخزين الاساسية عن طريق وحدة الإدخال مثل

INPUT, READ, PRINT.

2. تعليمات وأوامر تداول البيانات : وهي التعليمات التي تقوم بعمليات معالجة البيانات داخل الحاسوب مستخدمة وحدة الحساب والمنطق .

3. تعليمات واوامر التحكم : وتشمل التعليمات التي تؤدي الى تغير تسلسل تنفيذ جمل البرنامج نتيجة إجراء اختبار معين باستخدام وحدة الحساب والمنطق .

4. تعليمات وأوامر الإخراج : وهي التعليمات التي تقوم بإخراج البيانات المعالجة على شكل معلومات من الذاكرة الرئيسية الى وسط الإخراج

بواسطة وحدات الإخراج مثل : الطابعة الشاشة

5. تعليمات تعريفية : وهي التعليمات التي تقوم بتعريف الحاسوب على :

أ. مواضع التخزين الرئيسية .

ب. نوع البيانات التي يمكن أن تخزن في المواضع التخزينية وهي ثلاثة

أنواع: رقمية ، حرفية ، حرفية ورقمية .

جـ أنواع الوحدات المستخدمة في البرامج مثل نوع وحدات الإدخال والإخراج .

(4-1-7) قواعد اللغة :

وهي مجموعة القواعد والقيود التي يجب ان نلتزم بها عند كتابة البرنامج

حتى يستطيع الحاسوب تنفيذ تعليمات وأوامر البرامج المكتوبة بهذه اللغة .

(2 7) أنواع لغات البرمجة

تصنف لغات برمجة الحاسوب الى مستويين اساسيين : لغات المستوى الواطئ

ولغات المستوى العالي .

(1-2-7) لغات المستوى الواطئ

وتنقسم أيضا الى نوعين هما لغة الآلة و لغة التجميع :

1. لغة الآلة **Machine Language**: وهي اللغة الوحيدة التي يفهمها

الحاسوب مباشرة دون وسيط ، وتعليمات هذه اللغة هي مجموعة من الأرقام

الثنائية إما (1 , 0) ظهرت هذه اللغة مع بداية ظهور الحاسوب وكان

المبرمجون بلغة الآلة يحتاجون الى معرفة مكونات الحاسوب وامكانياته ، مما أدى

الى صعوبة فهم تلك اللغة وخاصة أن لكل حاسوب لغة آلة خاصة به ،

وبالتالي لا يمكن نقل البرنامج من حاسوب معين الى حاسوب آخر لذلك اقتصر استخدامها على الشركات الصانعة للحواسيب .

2. لغة التجميع او اللغة الرمزية : **Assembly Language** : وهي مرحلة متقدمة من لغة الآلة واسهل نسبياً منها ، وقد قامت الشركات المنتجة للحواسيب بتصميمها للتغلب على صعوبة استخدام لغة الآلة حتى تساعد على انتشار الحواسيب وقد تم استبدال الأرقام الثنائية برمز وهو عبارة عن حرفين أو ثلاثة أحرف يكون اسهل في تذكرها وكتابتها ويطلق عليها (Mnemonic) ، وتختلف الرموز المستخدمة باختلاف طراز لغة التجميع والشركة المنتجة له . وتعتبر هذه اللغة مرحلة وسط بين لغة الآلة واللغات ذات المستوى العالي ، وتحتاج لغة التجميع الى برنامج مترجم لترجمتها الى لغة الآلة التي يفهمها الحاسوب . ومن ابرز عيوب لغة التجميع هو ارتباطها بالآلة ، فكل آلة لها لغة التجميع الخاصة بها ، ويقصد بالآلة هنا تحديدا المعالج Processor او المعالج الدقيق Microprocessor .

(7 2 2) لغات المستوى العالي : High - level Languages

وقد سميت بهذا الاسم لبعدها عن لغة الآلة ، ولا يتطلب من المبرمجين المعرفة بالتفاصيل الدقيقة لجهاز الحاسوب كما هو الحال بالنسبة للغات ذات المستوى المنخفض و لكن يتطلب منه الامام العام بأسلوب عمل الحاسوب ومنطقه ، أما المعرفة التفصيلية فتكون لجوانب المشكلات والمسائل الخاصة ، بالتطبيقات ويمكن ايجاز مميزات اللغات ذات المستوى العالي فيما يلي :

1. عدم ارتباطها بآلة معينة مثل لغة الآلة ولغة التجميع .
2. سهولة نقلها وسهولة كتابة البرامج بها وذلك لاستخدامها كلمات وتعابير مشابهة لتلك التي يستخدمها الانسان في حياته العادية .
3. سهولة اكتشاف الأخطاء وتصحيحها .

4. لا تحتاج عملية تغيير الحاسوب بحاسوب آخر الى تغيير كبير في البرامج وذلك لان اللغة مصممة أساساً لحل نوعية معينة من المشاكل وليست لنوعية معينة من المعالجات .

5. توفير الوقت والجهد للمبرمجين مقارنة مع الوقت والجهد المبذول باستخدام لغات المستوى المنخفض .

6. سهولة التعامل مع الحاسوب .

وتنقسم لغات هذا المستوى الى عدة مستويات اخرى يمكن تصنيفها كما يلي :

- | | |
|--------------------------|-----------------------------------|
| 1 - لغات مرتبطة بالاسلوب | Procedure-Oriented Languages |
| 2 - لغات مرتبطة بالمشكلة | Problem-Oriented Languages |
| 3 - لغات استفسارية | Query Languages |
| 4 - لغات الذكاء الصناعي | Artificial Intelligence Languages |
| 5 - لغات البرمجة الشيئية | Object-Oriented Languages |

وفيما يلي نبذة عن اهم لغات كل مستوى من المستويات السابقة .

(1-2-2-7) لغات مرتبطة بالاسلوب

Procedure-Oriented Languages

هذه اللغات تتسم بالمرونة ويوجد منها لغات متعددة وتنقسم من حيث

الاستخدام الى :

1. لغات للتطبيقات التجارية وإدارة الأعمال
 2. لغات للتطبيقات العلمية
 3. لغات عامة الاغراض
- أولاً : لغات التطبيقات التجارية وإدارة الاعمال :
- 1) لغة الكوبول (Cobol)

اشتقت كلمة كوبول من الحروف الاولى للعبارة :

Common Business Oriented Language

وتعني اللغة الموجهة للأعمال التجارية العامة ، بدأ ظهورها نهاية عام 1959 بدعم من وزارة الدفاع الأمريكية . ثم تطورت في صور كثيرة حتى استقرت على آخر صورة لها عام 1985 (Cobol 85).

من مزايا لغة (Cobol) هي :

1. استقلالها عن جهاز الحاسوب اذ يمكن استخدامها على أنواع مختلفة من الحواسيب .
2. اعتمادها على أصول قياسية موحدة .
3. قدرتها على التحكم في البيانات ، و تنوع اساليب معالجتها .
4. قدرتها على تنظيم الملفات و توثيقها .
5. انتشارها الواسع في مجال التطبيقات التجارية .
6. يمكن استعمال اسماء و جمل و عبارات قد تصل طولها الى (30) حرفاً ، الأمر الذي تفتقد له لغة البيسك والفورتران .

اما عيوب لغة (Cobol) فهي :

1. طول برامجها مقارنة مع لغات اخرى ، وهذا قد يسبب الملل للمبرمج و فقدان السيطرة على اجزاء البرنامج .
2. قصورها عن برمجة المعادلات الرياضية و المشاكل الاقتصادية المعقدة .
3. بسبب تعاملها مع مجموعات كبيرة من الملفات والبيانات فانها تستهلك مساحة كبيرة من الذاكرة في الحاسوب .

(2) لغة آر بي جي (RPG)

وهذه اللغة اشتقت كلمة آر بي جي من الحروف الاولى للعبارة التالية :

Report Program Generator وتعني مولد برامج التقارير ، استحدثت هذه اللغة عام 1946 من قبل شركة IBM ثم طورها عام 1970 لحل مسائل مختلفة وخاصة التجارية منها ، وما يتعلق بتحديث الملفات ، إذ أن الأعمال التجارية

تحتاج دائماً الى إعداد الوثائق والشيكات والتقارير العادية وهذه اللغة صممت أساساً لتحقيق هذا الهدف بالإضافة الى انها لغة سهلة التعلم .

من عيوب لغة (RPG) :

1. لغة محدودة الانتشار فهي خاصة ببعض الحواسيب .
2. امكانياتها الرياضية محدودة جداً بالرغم من أنه هناك إصدارات جديدة من PRG تستطيع التعامل مع المصفوفات .

ثانياً : لغات التطبيقات العلمية :

1) لغة فورتران Fortran

وهي اختصار لـ (FORMula TRANslator) أي مترجم معادلات ، وهي لغة علمية هندسية من اقدم لغات المستوى العالي ، وقد وضعت أصولها شركة IBM عام 1957 .

من ميزات لغة (Fortran) أنها :

1. متخصصة في برمجة المسائل العلمية والهندسية .
2. ذات تعليمات واصطلاحات مختصرة وسهلة .
3. لغة متطورة ، واصدارتها الاخيرة فورتران 77 (Fortran 77) و Fortran 90 يضم اضافات وخصائص لم تكن موجودة سابقا ، حتى أنه اصبح بالامكان استخدامها لبرمجة المسائل التجارية ، وعمليات البرمجة الرمزية والحرفية .

اما عيوب لغة (Fortran) فهي :

1. تركيبها لا يسمح باستخدام البرمجة الهيكلية Structured Programming .
2. اعتمادها على ترميز الاسطر ضمن اعمدة محددة مما يجعلها لا تناسب اولئك الذين اعتادوا على البرامج الحرة في الشكل .

(2) لغة ألول Algol

اشتق اسمها من كلمتي ALGOrtimic Language أي لغة الخوارزميات، ظهرت عام 1958 وكانت تهدف الى حل المسائل العلمية والعقدية، سميت في ذلك الوقت Alogo 58 وتم تنقيحها عام 1960 ، وتتميز بالمنهجية ، والهيكلية والدقة في تعريف اللغة و تحديد اساليب تخطيط البرامج ، وهذه اللغة اكثر شيوعاً في اوروبا على عكس لغة الفورتران المنتشرة في امريكا اكثر من اوروبا بسبب سيطرة شركة IBM في امريكا و التي تفضل لغة الفورتران .

(3) لغة أبل APL

صممت عام 1962 من اجل العمل بنظام المشاركة الزمنية ، ولم يسمح لها بالاستخدام العام الا في عام 1968 ، و تعتبر لغة سهلة التعلم وتشتهر باستعمالها لأغراض المعالجة الجدولية Table Handling .

من عيوب لغة APL :

1. انها تستخدم حروفا لاتينية لبعض تعليماتها .
2. عدد تعليماتها كبير مقارنة مع لغات اخرى.
3. تحتاج الى مترجم كبير وذاكرة واسعة جدا في الحاسوب .

ثالثا : لغات عامة الاغراض :

(1) لغة بيسك Basic

كلمة Basic اختصارا للكلمات التالية:

وتعني

كود التعليمات الرمزية المتعددة الاغراض للمبتدئين ، وضعت أصول هذه اللغة في كلية دارتموث الأمريكية على يدي كل من جون كيمي وتوماس كيرتز عام 1965 وكان الهدف من وضعها ان تكون لغة تعليمية للمدارس والجامعات ، وفي اواخر

الستينات تطور استخدامها بسبب ادخال نظام المشاركة الزمنية الى بيئة الحاسوب فتعدت حدود المدارس والجامعات لتشمل الاعمال التجارية .

من أهم ميزات لغة (Basic) :

1. سهولة تعلمها وسرعة فهم رموزها واصطلاحاتها
2. تستخدم لاغراض متعددة منها التعليمية ومنها التجارية .
3. تطوير الشركات العالمية المنتجة للبرامج لها باستمرار ، وقد ظهرت عدة إصدارات للغة البيسك :

Basic - Gwbasic - Turbo Basic - Quick Basic - Visual Basic

4. توفرها في معظم الحواسيب الشخصية .
5. سرعة استجابة وتنفيذ البرامج عند ادخالها ، اذ ان هذه اللغة تعمل على نظام الترجمة المباشرة Interpretation وهو نظام يوفر فرصة التخاطب مع الحاسوب بحيث يتم اكتشاف اخطاء البرامج بسرعة و يتم اصلاحها .

2) لغة باسكال PASCAL

سميت هذه اللغة باسم عالم الرياضيات الفرنسي باسكال ، وقد صممت هذه اللغة في أواخر الستينات وأوائل السبعينات من قبل البروفيسور نيكولاس ويرث في المعهد الفيدرالي السويسري للتكنولوجيا ، وهي أول لغة رئيسية تم تصميمها بعد وضوح السمات الأساسية للبرمجة الهيكلية .

من مزايا لغة (Pascal) :

1. أنها تتفوق على اللغات الراقية الأخرى باعتمادها مفهوم البرمجة الهيكلية ، ذات الترتيب المنطقي المتسلسل .
2. يمكن استعمالها في المجالات الرياضية ، والعلمية والهندسية والتجارية والإدارية كما انها لغة تعليم جامعي .
3. تجمع بين مزايا الفورتران ومزايا البيسك ومزايا لغة الجوال .
4. برامجها قصيرة اذا ما قيس ببرنامج الكوبول في التطبيقات التجارية .

(3) لغة بي ال واحد (PL/1)

مشتقة من العبارة (Program Language / One) صممت هذه اللغة في منتصف الستينات في محاولة لتجميع خصائص معظم اللغات وخاصة الفورتران و الكوبول في لغة واحدة لدى شركة IBM الأمريكية ، وقد ظهرت لغة PL/1 القياسية طبقا للقياسات العالمية ANSE في عام 1976 ، وبعض الاصدارات من هذه اللغة القياسية اصبح متاحا على الحواسيب الشخصية .

من ميزات لغة (PL/1) :

1. سهولة تعلمها ، وسرعة كتابة البرامج بها لأنها لغة اشد اختصارا من غيرها .
2. تجمع بين الاغراض الهندسية و العملية و بين الاغراض التجارية .
3. تستعمل اسلوب البرمجة الهيكلية .

من عيوب لغة (PL/1) :

1. تكاد أن تكون لغة خاصة بشركة IBM ، فهي محدودة الانتشار .
2. تشبه الكوبول في حاجتها الى مساحات تخزينية واسعة في ذاكرة الحاسوب .

(4) لغة أدا ADA :

سميت هذه اللغة باسم اول مبرمجة وهي اجستا أدا بايرون ، صممت لغة أدا في وزارة الدفاع الأمريكية عام 1980 بهدف ان تكون لغة قياسية للأنظمة العسكرية ثم استخدمت بعد ذلك للاغراض التجارية ، وتعتبر لغة (أدا) لغة البرمجة الهيكلية تساعد في تصميم المنظومات الهندسية جزءا جزءا ، ولديها القدرة على فحص البرامج والتأكد من خلوها من الاخطاء قبل تنفيذها ، إلا انها تحتاج الى سعة ذاكرة كبيرة ، مما يجعل استخدامها على الحواسيب الصغيرة أمرا صعبا .

5) لغة سي C :

انتجت هذه اللغة في معامل بل في اوائل السبعينات واستخدمت لكتابة الانتاج الاول لنظام التشغيل يونيكس ، وتوجد اصدارات متعددة للغة الـ سي تعمل مع الحواسيب الشخصية والحواسيب الكبيرة والمتوسطة .

من ميزات لغة C :

1. تعتبر هذه اللغة وسطا بين اللغات الراقية كالبيسك والفورتران والكوبول والباسكال وبين لغة التجميع ، فهي تحل محل لغة التجميع في عملياتها ، وفي الوقت نفسه تؤدي اعمال اللغات الراقية ، لذلك انتشر استخدامها في الاوانة الاخيرة بشكل ملحوظ .
2. تعتبر من لغات البرمجة الهيكلية مثل الباسكال .
3. يمكن تحويل البرامج المكتوبة بلغة سي من نوع من الحواسيب الى نوع اخر دون مشقة .

(7-2 2 2) لغات مرتبطة بالمشكلة Problem - Oriented Languages

وهي لغات صممت لحل مشاكل معينة او تطبيقات خاصة ، وهي لا تتطلب عادة برمجة تفصيلية كاللغات المرتبطة بالاسلوب وانما يكون التركيز اكثر على المدخلات والمخرجات ومن امثلة تلك اللغات :

1) لغة لوجو LOGO :

تم إنتاجها في أواخر الستينات على يد سيمون بايت وزملائه في معهد (MET) بامريكا وقد صممت خصيصا للأطفال فهي لغة بسيطة وسهلة التعلم ويسهل بواسطتها عمل الرسوم وتلوينها وتحريكها وتستخدم أيضا في المدارس لتأليف المقاطع الموسيقية وإدارة البيانات والتحكم في النصوص ،بالإضافة الى انها تعمل على الحواسيب الشخصية .

(2) لغة بيلوت PILOT :

وهي اختصار للعبارة Programmed Inquiry Learning or Teaching أي الاستفسار و التعليم او التدريس المبرمج ، وقد صممت لكتابة البرامج التعليمية فيمكن بواسطتها كتابة الاسئلة ثم اجابتها ورد فعل الحاسوب في حالة الصواب أو الخطأ كما يمكن بسهولة عمل الرسومات المختلفة بها .

(3-2-2-7) لغات استفسارية Query Languages

صممت هذه اللغات أساسا للمدراء . وهي عبارة عن تعليمات باللغة الانكليزية تماما و لا تحتوي الا على عمليات حسابية محدودة وتتميز بقدرتها على عمل التقارير وترتيب واختيار البطاقات والسجلات بطريقة آلية وهو ما يهم المدراء.

(4-2-2-7) لغات الذكاء الاصطناعي

Artificial Intelligence Languages

(1) لغة ليسب LISP

وهي اختصار (List Programming Language) أي لغة برمجة القوائم ، صممها جون مكارثي (1959-1960) لتدعيم البحوث في مجال الذكاء الاصطناعي حيث للتحكم في البيانات غير الرقمية وقد ظلت حتى الان اللغة المفضلة في الولايات المتحدة الامريكية للباحثين في مجال الذكاء الاصطناعي.

(2) لغة برولوج PROLOG

كلمة Prolog اختصارا للعبارة (Programming in Logic) أي البرمجة المنطقية ، وقد تم تصميمها في فرنسا ، ثم اختارها اليابانيون لتكون اللغة القياسية للجيل الخامس للحواسيب (وهو المشروع الذي خططت له ونفذته اليابان لأنتاج حواسيب ذكية تحاكي الانسان في قدراته على التحليل والاستنتاج) ، بالاضافة الى فعاليتها في مجال النظم الخبيرة القادرة على تخزين المعرفة و الخبرات المختلفة .

(5-2-2-7) لغات البرمجة الشيئية

Object-Oriented Languages

وهي لغات تقوم على تعريف الأشياء المستخدمة بها ، و يكون تعريف الشيء مكونا من حالة الشيء والمنهج (Method) أي يوصف العمليات التي يمكن تنفيذها على هذا الشيء ، ومن امثلة لغات البرمجة الشيئية :

Smalltalk .1

C++ .2

Visual C++ .3

Pascal 5.5 .4

أسئلة وتمارين عامة

1. عرف اللغة البرمجية ؟
2. عدد مكونات اية لغة برمجية ؟
3. تكلم باختصار عن الفرق بين لغات المستوى العالي ولغات المستوى المنخفض ؟
4. عدد انواع لغات المستوى العالي مع ذكر اسم احدى لغات كل نوع ؟
5. اذكر أهم ميزات لغة باسكال ؟

الفصل الثامن

برمجة الحاسوب والمخططات الإنسيابية

Computer Programming

تعتبر برمجة الحاسوب علماً وفناً في نفس الوقت ، ولحل أي مشكلة باستخدام الحاسوب تتبع عدداً من الخطوات ، وحسب درجة كفاءة هذه الخطوات تتحدد كفاءة تشغيل المسألة حاسوبياً ، والجدير بالذكر ان اهم هذه الخطوات يتم تنفيذها خارج الحاسوب وبدون استخدام تقنيات الحاسوب ، إذ إنها تمثل منطق حل المسألة.

لذلك يمكن تقسيم عملية اعداد برنامج يتم تطبيقه على الحاسوب إلى ثلاث مراحل اساسية و كل مرحلة تتضمن مجموعة من الخطوات ، تؤدي جميعها بشكل متكامل الى الوصول الى برنامج حاسوبي للمشكلة المطلوب معالجتها باستخدام الحاسوب .

(8-1) المراحل الأساسية لحل أي مسألة حاسوبياً :

تتكون المراحل الأساسية لحل أية مسألة باستخدام الحاسوب من ثلاثة مراحل وهي كما يلي :

(8-1-1) المرحلة الاولى (دور الإنسان) :

إن دور الإنسان في هذه المرحلة تعتمد على المختص بالمشكلة وكذلك تعتمد على محلل النظم ومن ثم المبرمج وتتضمن هذه المرحلة الخطوات التالية :

الخطوة الأولى : تعريف المسألة وتحليلها

Problem Definition and Analysis

تعتبر هذه الخطوة من أهم وأصعب الخطوات على الإطلاق حيث يتم فيها تحديد أبعاد المسألة ، وتحديد الهدف المطلوب الوصول اليه وذلك بتحليل مفردات المسألة ووصفها ومن مجموعة من المهام وهي :

1. تحديد أهداف المسألة و تحديد النتائج المراد تحقيقها من حيث نوع المعلومات واسلوب عرضها وشكل المعلومات ، والتي تدعى المخرجات لنتمكن من تحديد المدخلات والعمليات اللازمة لتحقيقها ، فالنتائج تحدد وتعرف أولاً ونحصل عليها بعد إجراء المعالجة وإظهار النتائج .
 2. تحديد المدخلات اللازمة للحصول على المخرجات المعرفة مسبقاً، وكذلك تحديد شكل المدخلات ومواصفاتها بدقة .
 3. حصر طرق الحل المختلفة من وجهة نظر الحاسوب ، وتقييمها لاختيار الحل الأفضل والأكثر ملائمة للتنفيذ باستخدام الحاسوب من حيث سهولتها وسرعة تنفيذها والمساحة التي تحتاجها في ذاكرة الحاسوب .
- ويشارك في اعداد متطلبات هذه الخطوة الإحصائي في موضوع المشكلة المدروسة (المحاسب، المهندس ، ...) بالإضافة الى محلل النظم المختص بعلم الحاسوب لتهيئة المسألة ووضع أسلوب حل يتناسب مع منطق الحاسوب .

الخطوة الثانية : وضع خوارزمية الحل Algorithm

بعد اختيار الطريقة المثلى لحل المسألة قيد الدراسة في الخطوة السابقة نقوم بالتعبير عن أسلوب حل المسألة على شكل خطوات متسلسلة و متعاقبة و مترابطة منطقياً تؤدي الى الوصول الى حل المسألة. يدعى هذا التسلسل من الخطوات بالخوارزمية Algorithm ، و التي يمكن تعريفها كما يلي :

الخوارزمية (*): هي عدد من الأوامر أو التعليمات المترابطة والمرتببة على شكل خطوات متعاقبة، يمكن تنفيذ كل منها آلياً من أجل الوصول الى حل المسألة، وبشكل عام يمكن ان نطلق كلمة الخوارزمية على طريقة الحل اذا توفرت في هذه الخوارزمية النقاط التالية :

1. أن تحوي هذه الخوارزمية على عدد محدد من الخطوات للتوصل الى النتيجة، بمعنى آخر لا يمكن تسمية طريقة حل ما بالخوارزمية إذا حوت عدد غير منتهى من الخطوات .

2. يجب أن تكون الطريقة واضحة ، وتتضمن تعليمات مفصلة و صريحة و أن تكون كل خطوة واضحة بشكل لا لبس فيه .

3. أن تحوي الخوارزمية على طريقة لجميع الطريقة جميع الحالات الخاصة والتي يمكن ان تعترض الانسان خلال عملية الحل .

4. أن تكون الخوارزمية صالحة لحل جميع المسائل المشابه والتي تنطبق عليها نفس الشروط .

مثال (1) : اكتب خوارزمية ايجاد مساحة ومحيط دائرة نصف قطرها معلوم (R) الخوارزمية :

الخطوة (1) : ادخل قيمة نصف القطر R

الخطوة (2) : اجعل قيمة $PIE = 3.14$

الخطوة (3) : احسب مساحة الدائرة حسب العلاقة : $A = B * R * R$

الخطوة (4) : احسب محيط الدائرة حسب العلاقة : $C = 2 * B * R$

(*) : استخدمت الخوارزميات في الماضي بشكل واسع في اوربوا وامريكا ، وكانت تعني الوصف الدقيق لتنفيذ مهمة من المهمات أو حل مسألة من المسائل، وقد اشتق اسم هذه الكلمة من أسم العالم الشهير الخوارزمي (ابو جعفر محمد بن موسى الخوارزمي) وهو اول من استخدم العدد صفر ، و صاحب كتاب (الجبر و المقابلة) أما الآن فقد ارتبط هذا الاصطلاح ارتباطاً وثيقاً بعلم الحاسوب حتى أن احد اقطاب علم الحاسوب هو الاستاذ khuth يعرف علم الحاسوب على أنه (دراسة الخوارزميات) .

الخطوة (5) : اطبع قيم كل من A, C, R حيث تمثل R : نصف القطر ،
 A : المساحة ، C : المحيط .

مثال (2) : اكتب خوارزمية لتحديد فيما اذا كانت قيمة A مساوية لقيمة B ام
ان احدهما أكبر من الآخر وذلك باستخدام منطق الحاسوب .
الخوارزمية :

الخطوة الأولى : ادخل قيمة A, B

الخطوة الثانية : قارن A مع B ، اذا كانت $A = B$ عندئذ اطبع $A = B$
واذهب الى الخطوة الخامسة .

الخطوة الثالثة : اذا كانت $A < B$ عندئذ اطبع $A < B$ واذهب الى الخطوة
الخامسة .

الخطوة الرابعة : اطبع $A > B$.

الخطوة الخامسة : اما ادخال قيم جديدة لـ A, B والعودة الى الخطوة
الاولى او التوقف .

مثال (3) : ارادت شركة تجارية القيام بحساب الربح الصاف والذي يمثل NP
لتوزيعه على الشركاء في نهاية العام، وذلك بطرح المصاريف المختلفة E من الربح
الاجمالي GP ، فاذا كان عدد الشركاء N وهم متساوون في حصصهم هو N ،
فاكتب الخوارزمية المثالية لحل هذه المسألة .

الخوارزمية :

الخطوة الأولى : ادخال قيمة GP وقيمة E وقيمة N

الخطوة الثانية : احسب الربح الصافي من المعادلة : $NP = GP - E$

الخطوة الثالثة : احسب حصة كل شريك في الشركة من المعادلة $P = NP / N$

الخطوة الرابعة : اطبع النتيجة P, NP

الخطوة الثالثة : وضع المخطط الانسيابي Flowchart

ان المخطط الانسيابي^(*) (خريطة سير العمليات) هو التعبير عن الخوارزمية بالرسم وذلك باستخدام رموز واشكال اصطلاحية لمساعدة مخطط البرامج على وضع تصور مرئي لخطوات الحل اللازمة للمشكلة المدروسة بواسطة الحاسوب .
ولأهمية خرائط سير العمليات في حل المسائل الحاسوبية سنفرد فقرة مستقلة نتناول فيها انواع خرائط سير العمليات بشكل عام و خرائط سير العمليات المتعلقة بالبرامج بشكل خاص .

الخطوة الرابعة : كتابة البرنامج باحدى لغات البرمجة .

هذه الخطوة في الواقع تعتبر مرحلة مباشرة طالما المخطط التدفقي الانسيابي على مستوى عال من الدقة والتفصيل ، إذ يتم ترجمة الخوارزمية والمخطط التدفقي باستخدام احدى لغات تحرير البرامج، والتي يوجد العديد منها ، إلا أن السؤال الهام الذي يطرح نفسه : أي لغة برمجة يلزم استخدامها ؟

إن الإجابة على هذا السؤال ليس من السهولة كما يبدو للوهلة الاولى ولكن هناك العديد من الأسئلة التي تطرح في هذا المجال ، ومن خلال الاجابات عليها نستطيع تحديد اللغة المناسبة ، ونلخص هذه الاسئلة فيما يلي :

1. هل المبرمج لديه معرفة ودراية بهذه اللغة ؟ واذا لم يكن لديه الدراية فهل يمكنه تعلمها بسهولة ؟

2. هل استخدام هذه اللغة سيحقق أداء جيدا لهذه النوعية من التطبيقات ؟

3. هل هناك مترجم كفؤ لهذه اللغة ..؟ اذ أنه هناك فرق بين اللغة ومترجمها ، فاللغة هي مجموعة من القواعد والاصطلاحات تستخدم لتمثيل المعلومات من الانسان الى الحاسوب ، أما مترجم اللغة فهو عبارة عن برنامج كتب من قبل مبرمجين ، وظيفته تحويل البرنامج المكتوب بلغة معينة الى لغة الآلة .

(*) اول من استخدم المخططات الانسيابية بصفة رسمية هو جون فون نيومان عام 1945 ، وارتبطت منذ ظهورها بمجالات الحاسوب وما زال استخدامها مستمرا حتى الآن .

4. كيف يتم تشغيل التطبيق المطلوب ؟ إذ كان التطبيق سيتم تشغيله بصفة دورية ولمدة طويلة، عندئذ فان وقت التشغيل الأقل و حيز التخزين الاقل أكثر أهمية من ارتفاع تكاليف اعداد و تجهيز البرنامج التطبيقي و لذلك يفضل استخدام لغة التجميع Assembler مثلاً، من جهة اخرى اذا كان وقت استخدام التطبيق لفترة محدودة فانه من الأفضل أن يكون وقت تحرير البرامج أقل ما يمكن وفي هذه الحالة يفضل استخدام احدى لغات المستوى العالي السهلة .

5. هل سيتم تغيير وتعديل البرنامج دورياً ؟ إذ أن اللغات عالية المستوى أسهل في التعديل من اللغات ذات المستوى المنخفض وذلك لقرّبها من لغة الإنسان .

6. هل من المتوقع تغيير الأجهزة أو نظام التشغيل التي يعمل عليها البرنامج في المستقبل القريب ؟ إذ أنه ليست جميع اللغات متاح لها مترجمات لجميع اجهزة الحواسيب و نظم التشغيل ، كما أن هناك لغات يسهل تحويل البرامج المكتوبة بها من حاسوب الى آخر و من جيل حواسيب الى جيل آخر ..

وقد تعرضنا في الفصل السابق الى أنواع لغات البرمجة الحاسوبية وأهمية كل لغة ، وستعرض في الفصل القادم الى احدى لغات البرمجة بشكل مفصل (لغة باسكال) ، يدعى البرنامج المكتوب باحدى لغات البرمجة بالبرنامج المصدري Source Program.

(8-1-2) المرحلة الثانية دور الحاسوب :

ونقصد بدور الحاسوب استخدام التقنيات الحاسوبية المختلفة ، وتمر هذه المرحلة بعدة خطوات :

الخطوة الأولى : إدخال البرنامج إلى الحاسوب :

بعد الانتهاء من اعداد البرنامج ، فإنه يتعين إدخاله إلى الحاسوب للتأكد من صحة كتابته من جهة ، ثم لترجمته الى لغة الالة من جهة اخرى ، ويدعى البرنامج الجديد المترجم بالبرنامج الهدف Object Program وتتم الترجمة بواسطة برامج

خاصة تعدها الشركات المنتجة للحواسيب لترجمة البرامج المكتوبة بإحدى اللغات العالية المستوى الى لغة الآلة بهدف تحقيق مهمتين اساسيتين :

1. اكتشاف الاخطاء المعجمية و النحوية (وبعض الاخطاء المنطقية) .
2. ترجمة البرنامج الأولي (برنامج المصدر) المكتوب بإحدى اللغات العالية المستوى والخالي من الأخطاء إلى برنامج بلغة الآلة الذي يدعى بالبرنامج الهدف.

الخطوة الثانية : اختبار البرنامج وتصحيح اخطائه

Testing and Debugging

بعد الحصول على برنامج الهدف ، يتم تجربته للتأكد من صحته منطقياً وذلك بتنفيذه على عينة من البيانات الاختيارية (Test Data) فإذا ما ثبت صحة البرنامج منطقياً ، فإن ذلك يعني صحة طريقة الحل المقترحة . وبشكل عام تنقسم الأخطاء المحتملة وجودها في أي برنامج الى نوعين أساسيين :

1. أخطاء في قواعد اللغة (Syntax Errors) وهي أخطاء يكتشفها الحاسوب مباشرة لدى تشغيل البرنامج و يعطي الحاسوب رسالة خطأ بخصوصها ، ويمكن تصحيحها مباشرة بالعودة الى البرنامج المساعد help التابع للغة المستخدمة للتأكد من صحة استخدام وكتابة تعليمات اللغة .
2. أخطاء في تصميم البرنامج (Logical Errors) وهي أخطاء منطقية تؤدي الى نتائج خاطئة أو بعيدة عن المنطق و هذه الأخطاء قد تحدث رغم عدم وجود أخطاء في قواعد اللغة، ولا يمكن للحاسوب اكتشافها و بالتالي لايعطي أي رسائل خطأ بخصوصها و ينهي الحاسوب البرنامج نهاية طبيعية ، وهذا النوع من الأخطاء خطير جداً و يجب الإعداد الجيد له لاكتشافه وذلك باختبار البرنامج مرات عديدة على بيانات مختلفة تمثل الواقع الفعلي لظروف تشغيل البرنامج .

الخطوة الثالثة: اعداد البيانات والمعطيات وتنفيذ البرنامج Program Execution

حيث يتم في هذه الخطوة اعداد البيانات و المعطيات الحقيقية عن المشكلة المدروسة ليصار الى ادخالها حسب اسلوب الادخال المتبع في تصميم البرنامج ومن ثم معالجتها حسب تعليمات البرنامج الحاسوبي .

الخطوة الرابعة : الحصول على النتائج Results

في هذه الخطوة تقوم وحدات المخرجات (الطابعة ، الشاشة ، ...) بعرض النتائج ليصار الى دراستها وتحليلها ...

(8-1-3) المرحلة الثالثة دور الانسان : (التوثيق)

دور الإنسان في هذه المرحلة يتمثل بالمبرمج ومحلل النظم ، وتدعى هذه المرحلة بمرحلة توثيق البرنامج (Documentation) وهي عملية تجميع وتنظيم كافة التفاصيل و المعلومات المتعلقة بالبرنامج و كتابتها أو حفظها بأية وسيلة حفظ دائمة مثل الأقراص، الشرائط الممغنطة بحيث يمكن قراءة البيانات و المعلومات الموجودة بها عند الحاجة .

وتشمل هذه التفاصيل و المعلومات تسجيل الآتي :

1. تعريف المشكلة :ومن يستطيع الاستفادة من البرنامج وفي أي مجال يمكن استخدامه ، وما هي الأهداف التي يمكن تحقيقها منه ، و ...
2. وصف النظام :أي تحديد سمات المشكلة و نوع المدخلات المراد استخدامها وشكل ونوع المخرجات المطلوبة بكل وضوح ودقة .
3. وصف البرنامج : و يشمل خوارزمية الحل و خرائط سير العمليات و العينة الاختبارية للبيانات و نتائجها .
4. تعليمات التشغيل :وتشمل تحديد كيفية تحميل البرنامج و تشغيله و إيقافه بالتفاصيل ، و في مختلف ظروف التشغيل .

5. قائمة قواعد الأخطاء : و تشمل تعريف الاجراءات التي يتخذها مستخدم الحاسوب في حالة حدوث أخطاء معينة أو توقف جزء معين من البرنامج .

(8 - 2) مخططات سير العمليات (المخططات الانسيابية) Flowcharts

(8 - 2 - 1) مفهوم و أهمية مخططات سير العمليات :

كما ذكرنا سابقا إن الخوارزمية هي عبارة عن مجموعة من الخطوات المتسلسلة التي تصف بصورة مضبوطة وبدون أي غموض جميع الخطوات الرياضية و المنطقية اللازمة لحل مسألة ما، الا أنه في كثير من الاحيان يكون هذا الوصف معقدا و صعبا من حيث الامام بجوانب المسألة و تتبع مراحل حلها، لذلك فان خرائط سير العمليات تمثل وصفا تصويريا لخطوات الخوارزمية، وكثير من المبرمجين يفضلون خريطة سير العمليات عن الخوارزميات، اذ يمكن بواسطة الخريطة تتبع التسلسل المنطقي لحل المسألة بسهولة، وفي بعض الاحيان تكون عملية كتابة الخوارزميات اسهل بالاعتماد على خرائط سير العمليات من كتابة الخوارزمية مباشرة .

ولخرائط سير العمليات (المخططات الانسيابية) أهمية كبيرة نوجزها بما يلي:

1- تعطي تصور كامل عن خطوات حل المسألة في ذهن المبرمج تمكنه من الإحاطة الكاملة بكل أجزاء المسألة من بدايتها وحتى نهايتها .

2- تساعد المبرمج على تشخيص الأخطاء التي تقع عادة في البرامج وبخاصة الأخطاء المنطقية منها ، والتي يعتمد اكتشافها على اسلوب التسلسل المنطقي لخطوات حل المسألة لدى المبرمج .

3- تساعد المبرمج في ادخال أي تعديلات في أي جزء من أجزاء المسألة بسرعة ودون الحاجة لإعادة دراسة المسألة برمتها من جديد .

4- تساعد المبرمج على تتبع التسلسل المنطقي لخطوات الحل وخاصة في المسائل التي تكثر فيها الاحتمالات والتفرعات .

5- تعتبر رسوم خرائط سير العمليات المستعملة في تصميم حلول بعض المسائل مرجعا في حل مسائل أخرى مشابه ومفتاحا لحل مسائل جديدة لها علاقة مع المسائل القديمة المحولة .

(8-2-2) الاشكال والرموز الاصطلاحية المستخدمة في مخططات سير العمليات:
هناك رموز واصطلاحات متعارف عليها دوليا تستخدم عند اعداد مخطط سير العمليات لمشكلة معينة ، هذه الرموز والاصطلاحات جاءت نتيجة جهود المعهد القياسي الامريكي (American National Standard Institute) ANSI من هذه الرموز

تستخدم للبداية Start ، وللنهاية End



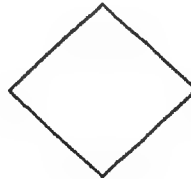
للدخال Input والاعراج Output
دون تحديد وسيلة الدخال والاعراج



عملية حسابية Calculation

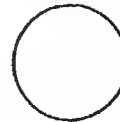


اتخاذ قرار منطقي (اختبار مقارنة)
Logical Decision Making



أسهم بين الاشكال والرموز الاصطلاحية تدل على اتجاه سير العمليات في خريطة سير العمليات flow lines

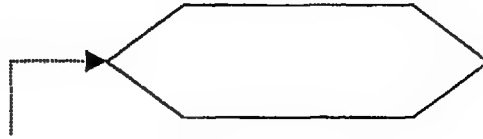
نقطة توصيل وربط بين أجزاء سير العمليات Connection



وثيقة اخراج التقرير على الطابعة Document



حلقة تكرارية Looping



استدعاء روتين (برنامج فرعي) Subroutine



إدخال وإخراج باستخدام أقراص مغناطيسية Magnetic Disk



بالإضافة إلى رموز أخرى لا ضرورة لذكرها بسبب استخدامها القليل جدا .

(3-2-8) انواع المخططات الانسيابية :

تنقسم المخططات الانسيابية (خرائط سير العمليات) بشكل عام الى

قسمين رئيسيين هما :

1) المخططات الانسيابية للأنظمة Systems Flowcharts

وهذا النوع يحدد بالرسوم و العلاقات المنطقية التي تربط بين مجموعة

العمليات المتتالية و المترابطة للنظام المطلوب ويتكون من عدد من الخرائط المهمة

كل منها توضح وظيفة أو مرحلة معينة داخل النظام من خلال النقاط التالية :

1. سير البيانات من حيث طريقة نقلها وتغذية الحاسوب بها وكذلك

تناولها ومعالجتها.

2. طريقة ربط برامج النظام بعضها ببعض .
 3. العلاقة بين الدوائر المستخدمة للحاسوب و النظام المقترح .
 4. تحديد شكل وطبيعة النتائج الخاصة بكل برنامج .
- وهناك أمثلة كثيرة على خرائط النظم منها ما تستخدمه الشركات والمؤسسات لوصف انظمتها المختلفة.

(2) المخططات الانسيابية للبرامج Programs Flowcharts

هذا النوع يحدد بالرسوم العلاقات المنطقية و التسلسل العام بين مجموعة من الاحداث والعمليات المترابطة والمتتالية التي يتكون منها البرنامج .
و هذا النوع من المخططات له أهمية بالغة في كتابة البرامج الحاسوبية ، وستناولها في الفقرات اللاحقة بالتفصيل .

(3-8) انواع المخططات الانسيابية للبرامج الحاسوبية :

يمكن تصنيف المخططات المتعلقة بالبرامج الى ثلاثة انواع :

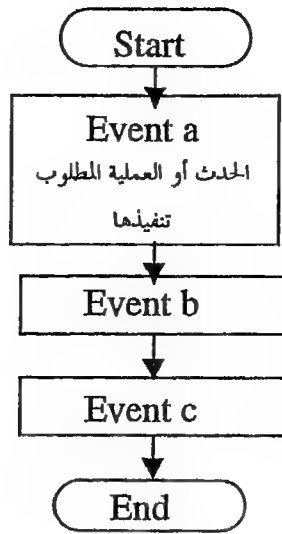
- 1 - مخططات التتابع البسيط Simple Sequential Flowcharts
- 2 - مخططات التفرع Branched Flowcharts
- 3 - مخططات التكرار Loop Flowcharts

ويمكن للبرنامج الواحد ان يشمل على أكثر من نوع واحد من هذه الانواع .

(1-3-8) مخططات التتابع البسيط

Simple Sequential Flowcharts

هذا النوع من المخططات يخلو من التفرعات والتكرارات او أي اختبار منطقي والشكل العام للمخطط الانسيابي كما في الشكل (3)

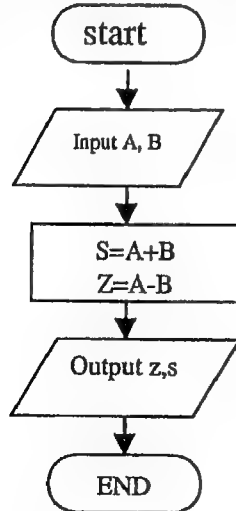


الشكل (3)

حيث : Event الواردة في الشكل السابق تعني الحادث أو العملية المطلوب تنفيذها إما عملية حسابية أو عملية إدخال أو إخراج .

مثال (4) : ارسـم مخطط سير برنامج لإيجاد مجموع عددين ما وحاصل طرحها .

خريطة سير العمليات الخوارزمية

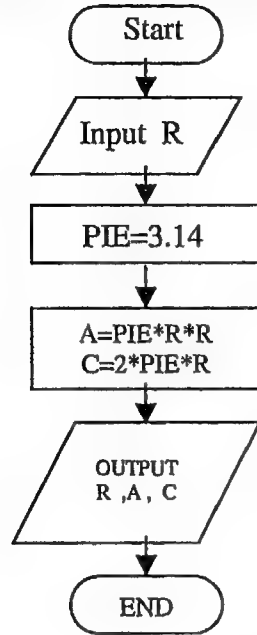


1. ابدأ
2. اقرأ القيم A, B
3. احسب قيمة S
 $S = A + B$
4. احسب قيمة Z
 $Z = A - B$
5. اطبع قيمة Z, S
6. توقف

الشكل (4)

مثال (5) : اكتب خوارزمية وارسم مخطط سير العمليات لبرنامج حساب مساحة ومحيط دائرة نصف قطرها معلوم R ؟

خريطة سير العمليات الخوارزمية



1. ابدأ
2. اقرأ قيمة R
3. اجعل $PIE = 3.14$
4. احسب المساحة A حسب العلاقة :
 $A = PIE * R * R$
5. احسب محيط C حسب العلاقة :
 $C = 2 * PIE * R$
6. اطبع قيم كل من R, C, A
7. توقف

الشكل (5)

(8 3 2) مخططات التفرع (الاختيار)

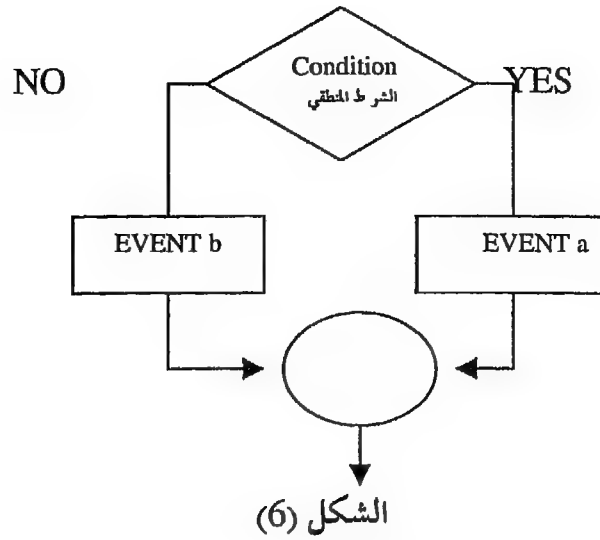
يتضمن هذا النوع من المخططات اتخاذ قرار أو مفاضلة بين خيارين أو أكثر

وهناك شكلين لمخططات الاختيار :

الشكل الاول : انظر الشكل (6)

في هذا الشكل اذا كان جواب الشرط المنطقي (Condition) نعم (Yes) فان الحدث التالي في التنفيذ هو (a)، اما اذا كان جواب الشرط بلا (No) فان الحدث التالي في التنفيذ هو (b) .

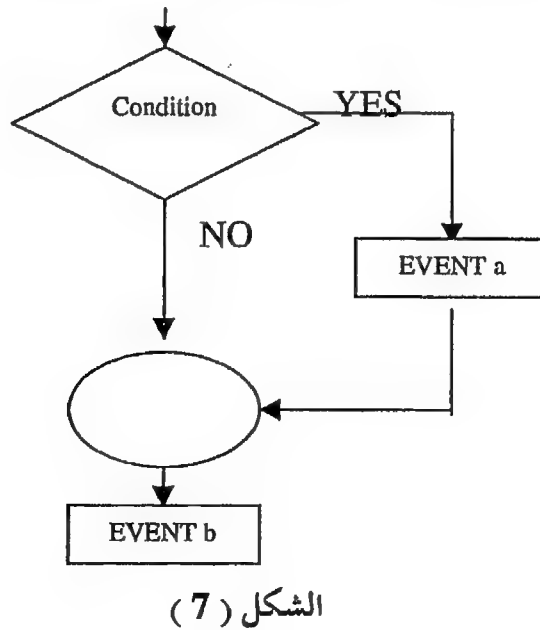
ويعرف هذا الشكل بصيغة اذا المزوجة أي IF - THEN - ELES



الشكل الثاني : انظر الشكل (7)

في هذا الشكل إذا كان جواب الشرط المنطقي (Condition) نعم (Yes) فان الحدث التالي في التنفيذ هو (a)، أما إذا كان جواب الشرط بلا (No) فان الحدث التالي في التنفيذ هو (b) .

ويعرف هذا الشكل بصيغة اذا البسيطة أي IF - THEN - STATEMENT



مثال (6) : اكتب خوارزمية الحل و ارسـم مخطط سير العمليات لقراءة قيمتين معلومتين واختيار اكبرهما باستخدام الشكل الاول و الشكل الثاني .
الخوارزمية باستخدام الشكل الاول :

1. ابدأ

2. ادخل قيمة كل من A , B

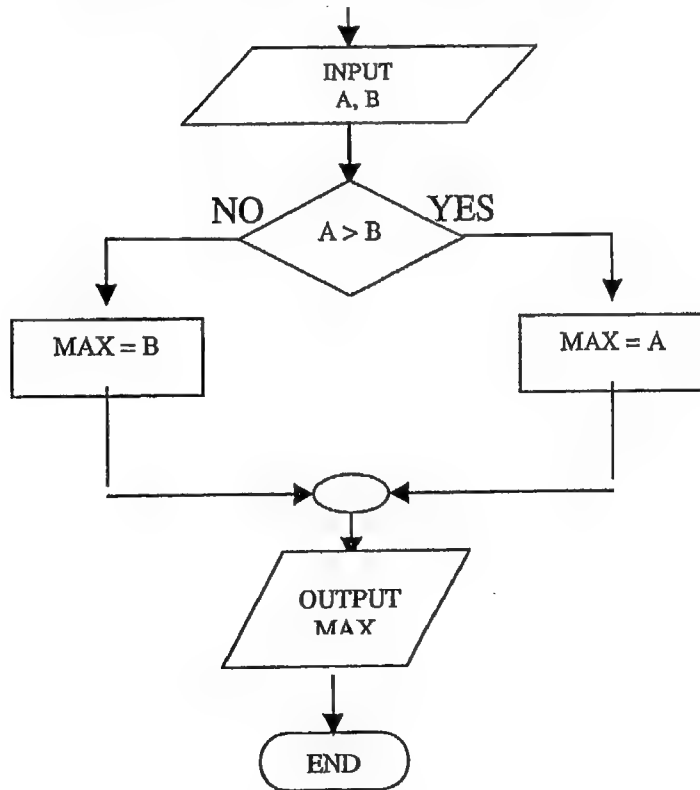
3. اذا كانت A اكبر من B : عندئذ اجعل : $Max = A$

والا اجعل : $Max = B$

4. اطبع النتيجة Max

5. توقف

والشكل (8) يبين خريطة سير العمليات لهذه الخوارزمية .



الشكل (8)

باستخدام الشكل الثاني :

الخوارزمية :

1. ابدأ

2. ادخل قيمة كل من A , B

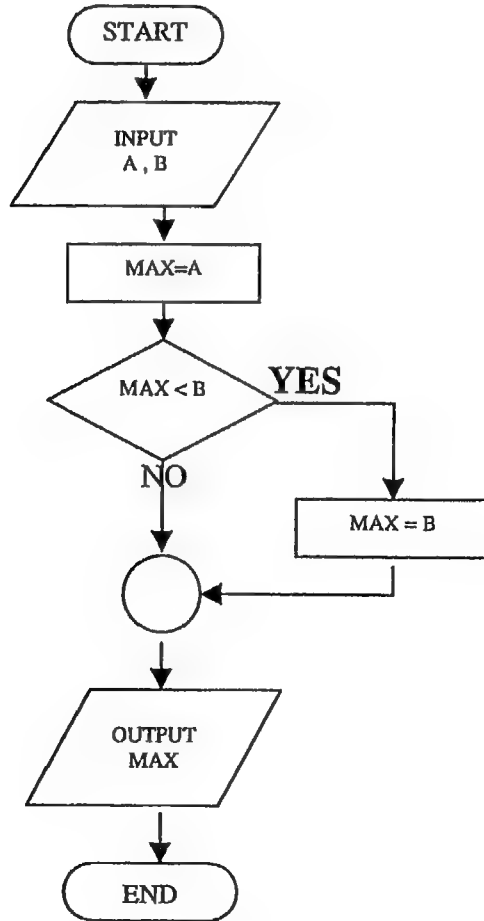
3. اجعل $Max = A$

4. اذا كانت $Max < B$ عندئذ اجعل $Max = B$

5. اطبع Max

6. توقف

والشكل (9) يبين سير العمليات لهذه الخوارزمية .



الشكل (9)

مثال (7) : اكتب خوارزمية وارسم مخطط سير العمليات لاعادة ترتيب قيمتين
ترتيباً تصاعدياً.

الخوارزمية :

1. ابدأ

2. ادخل القيمتين A , B

3. اذا كانت $A > B$ عندئذ استبدل بالشكل التالي :

اجعل $Z = A$

اجعل $A = B$

اجعل $B = Z$

4. اطبع A , B

5. توقف

والشكل (10) يبين خريطة سير العمليات لهذه الخوارزمية .
ملاحظة :بالإضافة الى الشكلين (9) و (10) لمخططات الاختيار هناك مخططات
يمكن ان تحتوي الشكلين معا و تدعى مخططات اذا المتداخلة Nested IF
Statement والمثالين التاليين يوضحان ذلك :

مثال (8) : اكتب خوارزمية الحل و ارسم مخطط سير العمليات للمشكلة التالية :
يقدم احد المخازن خصما على مبيعاته حسب ما يلي :

1- 20% اذا كانت طلبية الزبون اكثر من 200 دينار

2- 15% اذا كانت طلبية الزبون بين 100 دينار و اقل من 200 دينار

3- بدون خصم على الطلبية التي قيمتها أقل من 100 دينار

الخوارزمية :

1. ابدأ

2. أدخل قيمة طلبية الزبون S

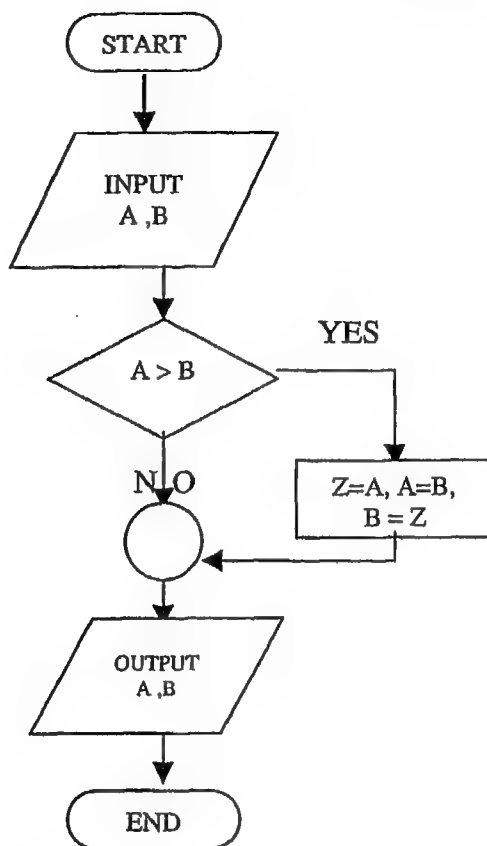
3. اذا كانت $S > 200$ احسب قيمة D حسب العلاقة التالية :

$$D = S * 0.20$$

والا اذا كانت $200 > S > 100$ احسب قيمة D حسب العلاقة التالية:

$$D = S * 0.15$$

و إلا فإن $D = 0$



الشكل (10)

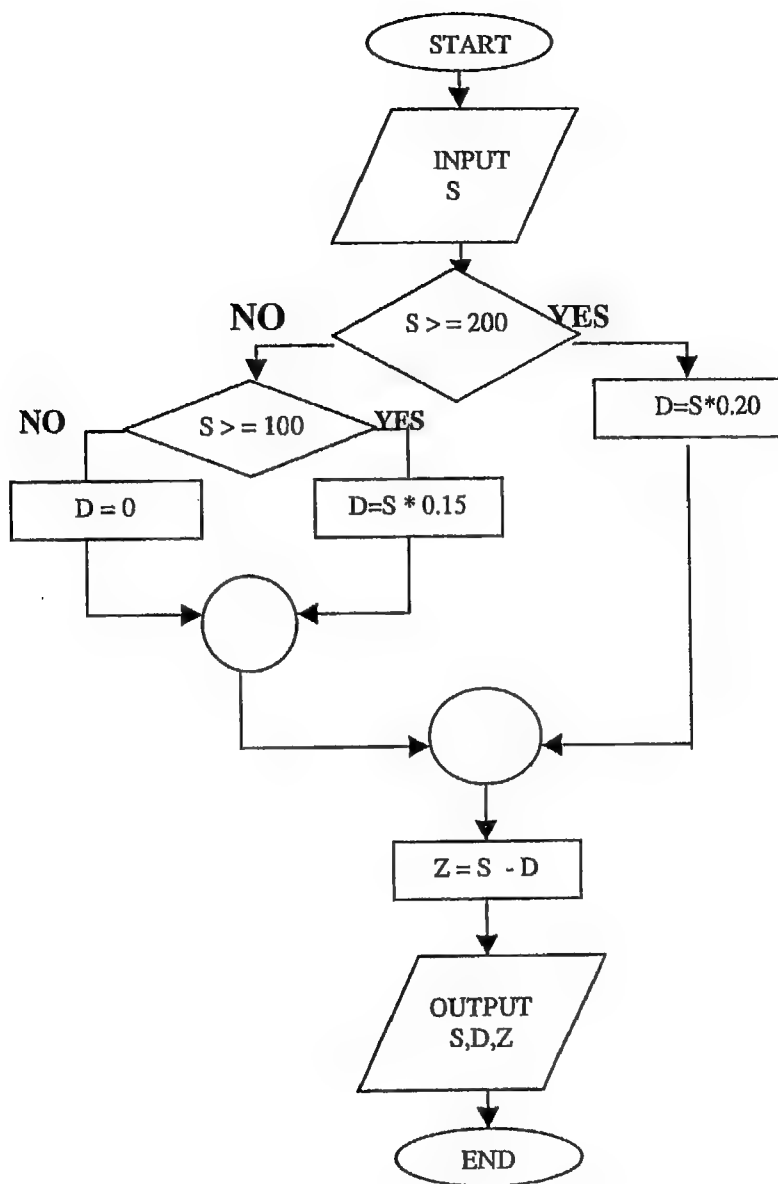
4. قيمة الطلبية النهائي :

$$Z = S - D$$

5. اطبع قيمة S , D , Z

6. توقف

والشكل (11) يبين مخطط سير العمليات لهذه الخوارزمية .



الشكل (11)

مثال (9) ارسم مخطط سير العمليات وخوارزمية الحل لحساب الدالة $F(X, Y)$ المعرفة كما يلي :

$$F(X, Y) = \begin{cases} X + Y & ; & X > Y \\ X^2 + 5 & ; & X = Y \\ Y + 2X & ; & X < Y \end{cases}$$

حيث قيم X, Y معلومتين

الخوارزمية :

-1 ابدأ

-2 إذا كانت $X > Y$ احسب :

$$F = X + Y$$

اما اذا كانت $X = Y$ عندئذ احسب العلاقة :

$$F = X * X + 5$$

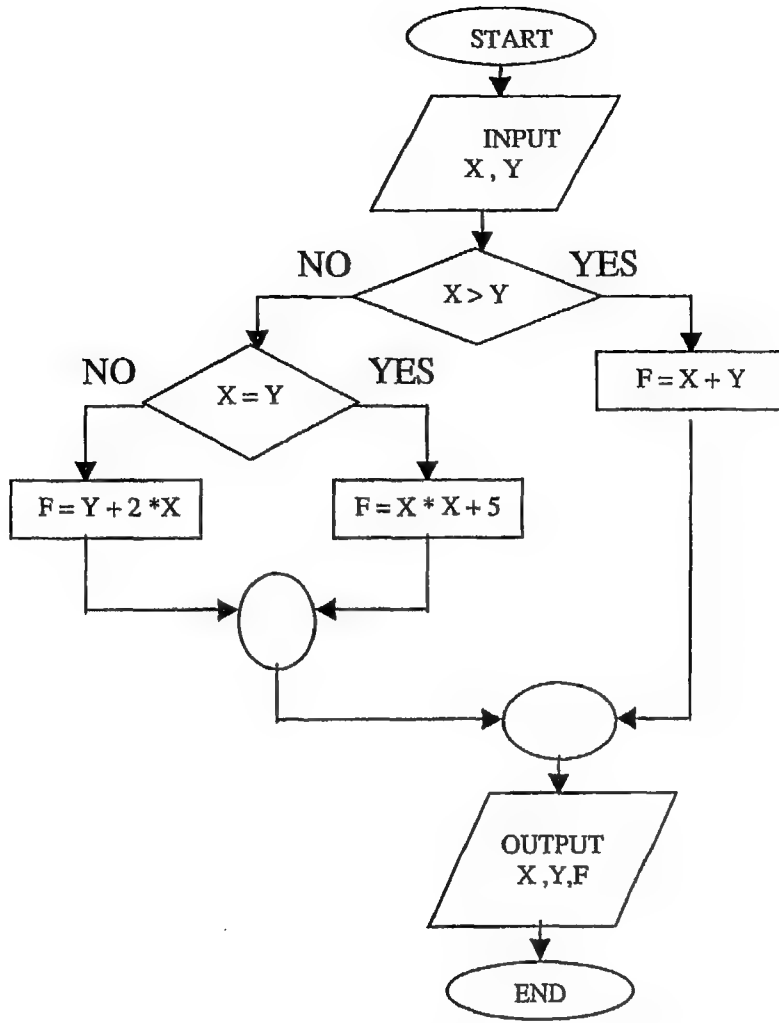
و إلا فاحسب :

$$F = Y + 2 * X$$

-3 اطبع قيمة كل من X, Y, F

-4 توقف

والشكل (12) يوضح مخطط سير العمليات لهذه الخوارزمية .

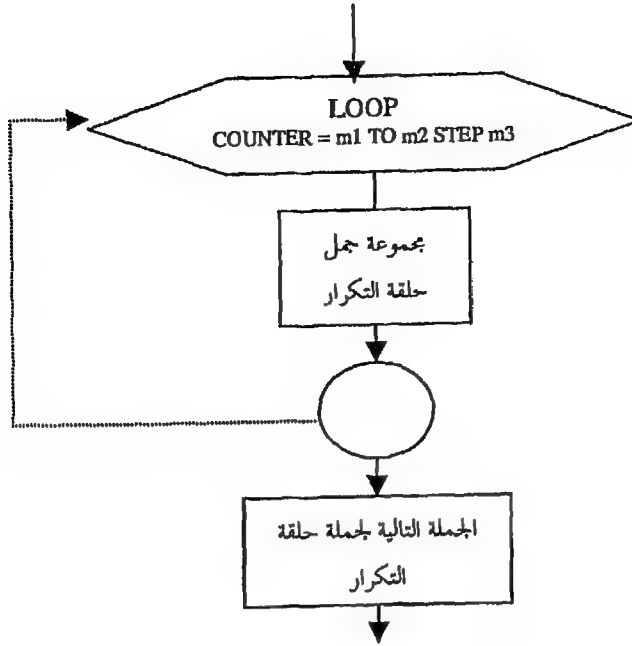


الشكل (12)

(8 3 3) مخططات التكرار :

و ينقسم هذا النوع من الخرائط الى نوعين :

- 1- اما تكرار تنفيذ مجموعة من الجمل عددا محدودا من المرات و يدعى في هذه الحالة بالتكرار غير المشروط ويأخذ الشكل العام التالي (انظر الشكل (13))



الشكل (13)

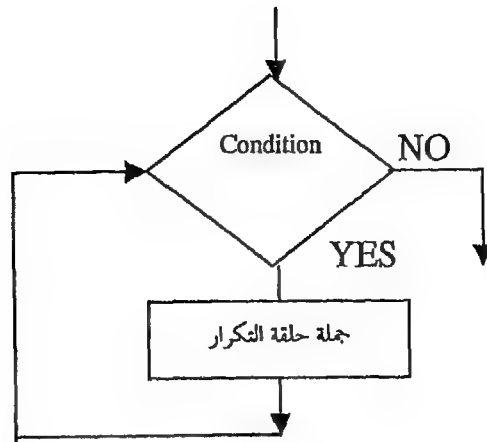
حيث : Counter اسم عداد، $m1$ القيمة الأولية للعداد، $m2$ القيمة النهائية للعداد، $m3$ مقدار التزايد .

ويتم تكرار مجموعة جمل حلقة التكرار عددا من المرات ابتداء من القيمة الابتدائية الأولية $m1$ و حتى القيمة النهائية $m2$ بزيادة مقدارها $m3$

2- أو تكرار جمل حلقة التكرار الى ان يتحقق شرط معين ويدعى في هذه الحالة بالتكرار المشروط ويأخذ أحد الشكلين التاليين :

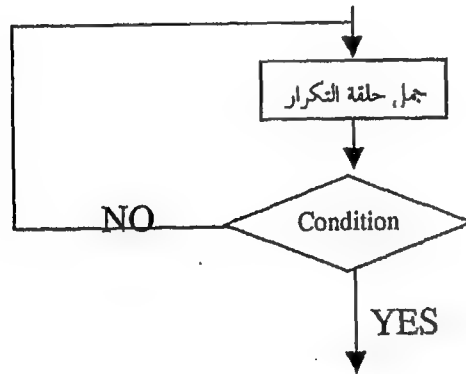
الشكل الأول : الشكل (14) ، في هذه الحالة يتكرر تنفيذ جمل حلقة التكرار ما دام الشرط محققا و تعرف هذه الحالة في اغلب لغات البرمجة بـ :

WHILE DO



الشكل (14)

الشكل الثاني : الشكل (15) ، في هذه الحالة يتكرر تنفيذ جمل حلقة التكرار حتى



الشكل (15)

يتحقق الشرط و تعرف هذه الحالة بـ :

REPEAT UNTIL

و قبل أن نعطي أمثله على هذا النوع من الخرائط لابد من التطرق الى مفهومي اساسين يستخدمان بشكل واسع لدى كتابة البرامج ورسم الخرائط وهما مفهومي العداد و المجموع .

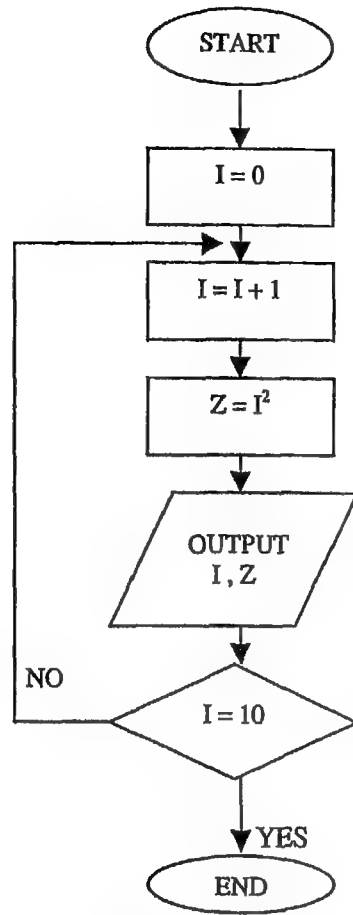
اولا : مفهوم العداد Counter

قد نحتاج في كثير من الاحيان الى ان نعد ظاهرة معينة عدد من المرات، مثلا اذا طلب ان نعد كل من الطلاب والطالبات في الصف، ان هذه العملية بالنسبة للانسان بديهيه وضمن قدراته العقلية التي يكتسبها منذ طفولته، إلا أن الحاسوب يحتاج الى اسلوب معين او ما يسمى بتصميم خوارزمية للعد Counting Algorithm و التي تتضمن الخطوات التالية :

1. اجعل العداد يبدأ بقيمة ابتدائية تساوي الصفر $Counter = 0$
 2. اجعل قيمة العداد تساوي القيمة السابقة مضافا اليها الزيادة الدورية m :
 $Counter = Counter + m$
 3. نفذ التعليمات والاوامر المراد تكرارها.
 1. اعد تكرار تنفيذ الخطوات (2) و (3) عددا من المرات حسب حالة التكرار .
- مثال (10) اكتب خوارزمية وارسم خريطة سير العمليات لطباعة الاعداد الطبيعية من 1 الى 10 و مربعاتها :
- * الخوارزمية باستخدام التكرار المشروط :

1. ابدأ
2. اجعل $I = 0$
3. اجعل $I = I + 1$
4. احسب Z من العلاقة التالية : $Z = I * I$
5. اطبع I, Z
6. هل $I = 10$ ، اذا كان الجواب No ، اعد تكرار الخطوات (3 و 4 و 5 و 6)
7. توقف

والشكل (16) يبين مخطط سير العمليات لهذه الحالة .



الشكل (16)

• الخوارزمية باستخدام التكرار الغير المشروط :

1. ابدأ

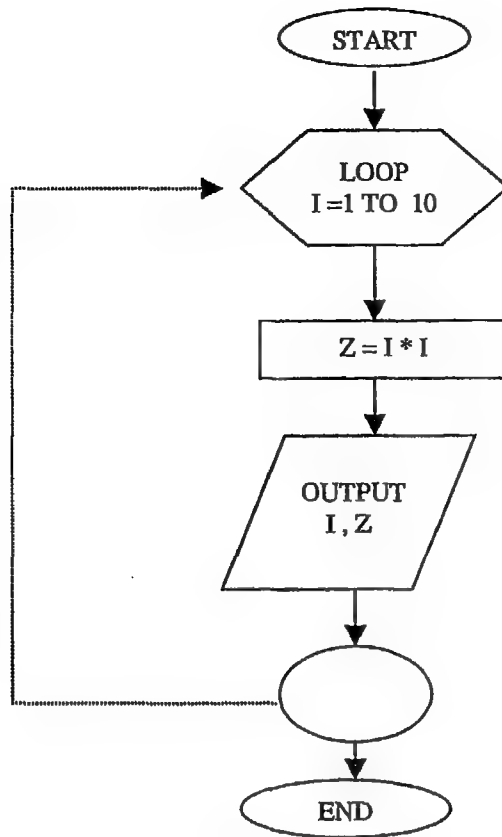
2. كرر مايلي عشرة مرات

- احسب قيمة Z حيث : $Z = I * I$

- اطبع قيمة I, Z

3. توقف

والشكل (17) يبين مخطط سير العمليات لهذه الخوارزمية .



الشكل (17)

ثانيا : مفهوم المجموع Summer

قد نحتاج في بعض الاحيان الى جمع مجموعة كبيرة من الاعداد التي تمثل معطيات ظاهرة معينة ، فمثلا ، كأن ترغب في ايجاد الوسط الحسابي لاعداد طلاب السنة الاولى في كلية الاقتصاد، ولتحقيق هذا يجب ان نحسب مجموع اعمار الطلاب ، أي اننا بحاجة الى ايجاد مجموع عدد معلوم من القيم ، وهنا نشير الى أنه من الناحية العملية لا يمكن إعطاء رمزا أبجديا لكل عدد يمثل عمر طالب إذ أنه في هذه الحالة نحتاج إلى مئات من الرموز ، و للتغلب على هذه المشكلة نستخدم ما نسمى بخوارزمية التجميع (Summers Algorithm) ، ونحتاج لاستخدام

هذه الخوارزمية متغيرين اثنين فقط إحداهما يمثل المتغير الذي نجميعه والآخر يمثل المجموع الاجمالي ، و الخوارزمية كما يلي :

1- اجعل قيمة المجموع الأولية تساوي الصفر $Sum = 0$

(ويمكن جعلها أي قيمة حسب المشكلة المدروسة)

2- اقرا القيمة التالية من القائمة ولتكن x

1- اجعل القيمة الجديدة للمجموع تساوي القيمة السابقة له مضافا اليها القيمة

المقروءة أي أن : $Sum = Sum + x$

2- اعد تنفيذ الخطوات (2 , 3) حسب حالة التكرار .

مثال (11) : اكتب خوارزمية و ارسم مخطط سير العمليات لاييجاد الوسط الحسابي لاعداد طلاب قسم الحاسوب .

* الخوارزمية باستخدام التكرار المشروط :

1. ابدأ

2. اجعل العداد $I = 0$ و المجموع $Sum = 0$

3. ادخل عدد الطلاب N

4. اضف واحد الى العداد I أي $I = I + 1$

5. ادخل عمر الطالب التالي Age

6. اجعل $Sum = Sum + Age$

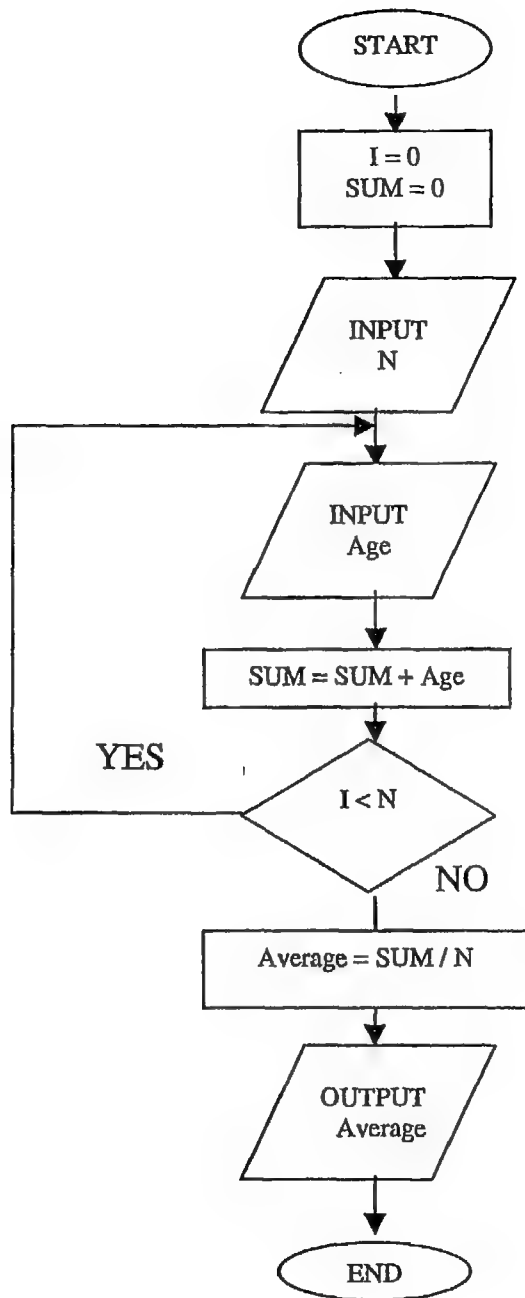
7. اذا كان $I < N$ عد الى الخطوة (4) والا تابع ،

8. احسب الوسط الحسابي :

$Average = Sum / N$

9. اطبع قيمة $Average$. 10. توقف

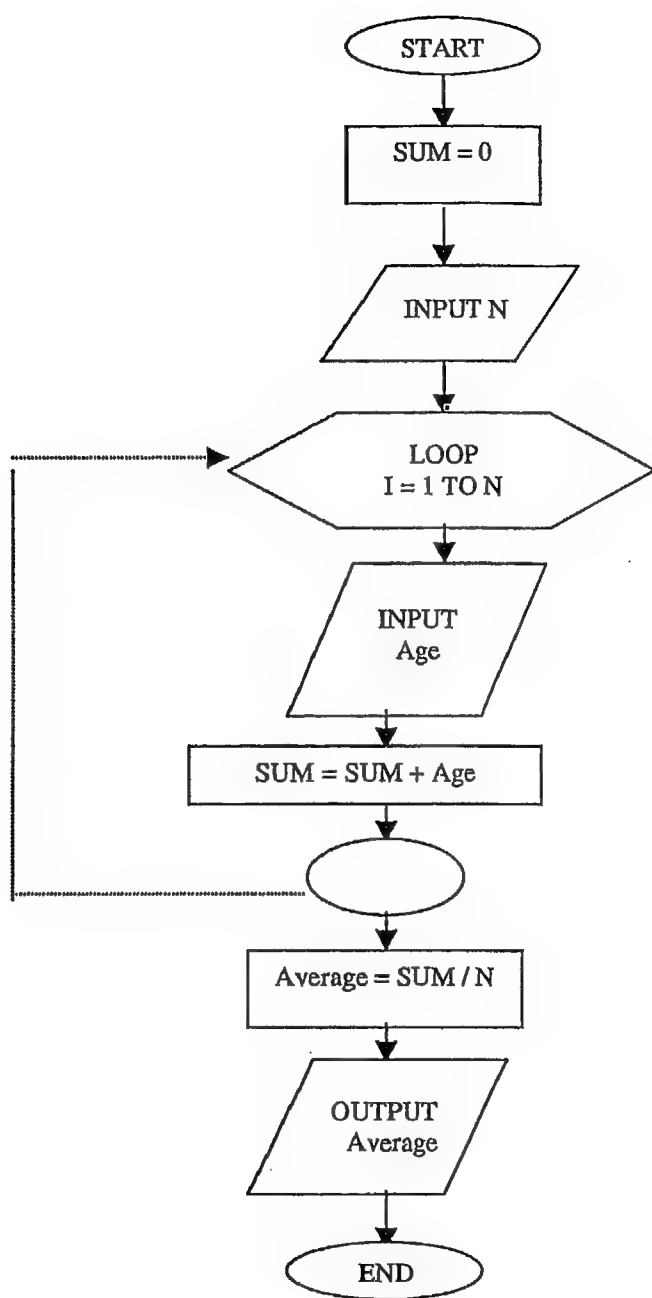
والشكل (18) يوضح مخطط سير العمليات لهذه الخوارزمية .



الشكل (18)

الخوارزمية باستخدام التكرار غير المشروط :

- 1- ابدأ .
 - 2- اجعل $SUM = 0$.
 - 3- اقرأ عدد الطلاب N .
 - 4- كرر مايلي N مرة .
 - إقرأ عمر الطالب التالي Age .
 - أجعل $sum = sum + Age$
 - 5- أحسب الوسط الحسابي :
 - $Average = sum / n$
 - 6- اطبع قيمة $Average$
 - 7- توقف .
- والشكل (19) يوضح مخطط سير العمليات لهذه الخوارزمية .



الشكل (19)

تمارين وأسئلة عامة

- (1) عدد المراحل الأساسية لحل مسألة حاسوبيا ؟
- (2) عرف ما يلي : الخوارزمية المخطط الانسيابي لسير العمليات البرنامج الهدف البرنامج المصدري .
- (3) اذكر بعض فوائد مخططات سير العمليات مع الشرح ؟
- (4) عدد انواع مخططات سير العمليات مع الشرح ؟
- (5) اكتب خوارزمية الحل و ارسم مخطط سير العمليات لكل من المشاكل التالية :
 - 1 - تحديد العدد الأكبر من بين ثلاثة اعداد A, B, C
 - 2 - تحديد العدد الأصغر من بين ثلاثة اعداد A, B, C
 - 3 - إيجاد جذور معادلة من الدرجة الثانية من الشكل $AX^2 + BX + C = 0$
 - 4 - إيجاد مضروب عدد حسب القاعدة التالية :

$$N! = \begin{cases} 1 & ; N = 0 \\ 1*2*3*4*.....*N & ; N > 0 \end{cases}$$
 - 5 - حساب قيمة الدالة : $Y = 2N^2$
- لجميع قيم X تقع في المجال $[-2, 2]$ و يتزايد مقداره 0.5
- 6 - حل جملة معادلتين بمجهولين .
- 7 - جمع مصفوفتين .
- 8 - ضرب مصفوفتين .
- 9 - حل جملة معادلات باستخدام طريقة تكرار .

الفصل التاسع

البرمجة بلغة باسكال

PASCAL PROGRAMMING

(اساسيات ومكونات — جمل الإدخال والإخراج — جمل التحكم والتكرار)

(1—9) المقدمة

تتصف لغة Pascal بأنها ذات أغراض متعددة ومختلفة في مجالات عدة، هندسية، وتجارية، وعلمية، لهذا فقد اخترناها لتكون اللغة المفضلة للجميع وهذه اللغة قد تم تطويرها أساساً من لغة Algol والتي وضعت عام 1960 وسميت في حينه البرمجة البنائية، وفي أوائل السبعينات طورها العالم (نيكلوس ويرث) وهو من جامعة زيوريخ للتكنولوجيا سويسرا- وسماها بلغة Pascal نسبة إلى العالم الفرنسي Pascal الذي اخترع أول حاسبة ميكانيكية في منتصف القرن السادس عشرة. إن لغة باسكال تطورت تطوراً سريعاً وأصبح لها عدة أشكال قياسية منها ISO /ANS1 والتي تبنتها مؤسسة المواصفات الدولية وكذلك معهد المواصفات الوطني الأمريكي، ثم بعد ذلك تطورت هذه اللغة وانتشر استخدامها انتشاراً واسعاً وبدأ استخدام النوع المطور منها والمسمى Turbo-Pascal، إن السبب في انتشار وتطور هذه اللغة يعود إلى سهولتها وقوتها البنائية ولاستخداماتها المتعددة وملائمتها للاستخدام في مختلف أنواع الحواسيب وكذلك ثقلها بمجموعة من اللغات الأقدم منها مثل Algol, Cobol, Basic, Fortran وسرعة إعطائها النتائج قياساً إلى اللغات الأخرى بالإضافة إلى كونها لغة تدريبية عالية المستوى .

هذا بالإضافة إلى ان هذه اللغة ركزت تركيزاً مباشراً على مفهوم البرمجة الهيكلية Structured – Programming والتي تسمى في بعض الأحيان البرمجة من الأعلى إلى الأسفل Top – Down Programming اي لغة التركيب المنطقي المتسلسل . لهذا فإن برنامج لغة باسكال يمكن أن يتكون من الأجزاء التالية:

- أ. الجزء الأول : وهو الجزء المسؤول عن كتابة وذكر أسم البرنامج والذي يأتي بعد كلمة PROGRAM ومن الممكن أن يتكون من سطر واحد أو أكثر.
 - ب. الجزء الثاني : وهو الجزء المسؤول عن التوضيحات والتعريفات للبرنامج وهو يحوي جملة واحدة أو أكثر من جمل التوضيح والإعلان عن جميع العناوين والثوابت والمتغيرات والإجراءات والإقترانات في البرنامج .
 - جـ الجزء الثالث : وهو الجزء الفعّال والمهم في البرنامج والذي يضم جمل باسكال الحسابية والمنطقية والذي يعتبر الجزء الرئيسي لبرنامج Pascal .
- ويمكن من خلال برنامج باسكال البسيط التالي توضيح الأجزاء المذكورة اعلاه وعلى الشكل التالي :

PROGRAM Prog1(Input,Output);	} الجزء الأول
(* Program to Compute and to Print the sum of two given number *)	
VAR	
a,b, Sum, : INTEGER ;	} الجزء الثاني
BEGIN	} الجزء الثالث
WRITE ('Enter the value of a : ');	
READ (a) ;	
WRITE ('Enter the Value of b : ');	
READ (b) ;	
Sum := a+b ;	
Writeln ('Sum = ', sum) ;	
END.	

وقبل الخوض في تفاصيل البرمجة الهيكلية وتعليماتها ومكوناتها لا بد من التعرف على أساسيات هذه اللغة .

(9 - 2) أساسيات ومكونات لغة باسكال

Pascal Fundamentals and Symbols

إن لغة باسكال تتكون من مجموعة من الرموز والأرقام والمسميات والكلمات والأسماء التعريفية ، ويمكن تقسيم لغة Pascal إلى :

(9-2-1) مجموعة رموز باسكال Pascal Character Set

إن مجموعة رموز لغة باسكال تنقسم إلى ثلاث أنواع هي :

أ. مجموعة الأرقام العشرية (Numeric digits) والتي تسمى في كثير من الأحيان أرقام اللغة العربية العشرية وهي عشرة أرقام هي :

0,1,2,3,4,5,6,7,8,9

ب. مجموعة الحروف اللاتينية أو (أحرف اللغة الإنجليزية) والتي تتكون من 26 حرفاً وهي (Letters)

A, B, C, D, E, F, G, H, M, N, L, P, Q, R, ...

بنوعيتها الحروف الكبيرة (Capital Letters) والحروف الصغيرة (Small

Letters)

جـ مجموعة الرموز الخاصة (Special Characters) والتي تشمل العمليات

الأربع والأشعار المنطقية وبعض الاشارات والعلامات الخاصة كما يلي :

)	>	,	+
(>=	:	-
] أو .)	>		*
[أو .(<=	;	/
} أو *)	<>	/	:=
{ أو *		@ أو &	=

(2-2-9) الكلمات المحجوزة (Reserved Words)

هذه الكلمات حوالي 39 كلمة تستخدمها لغة باسكال والتي تعتبر كلمات محجوزة قياسية معروفة مسبقاً لدى الحاسوب وتكتب هذه الكلمات علدة بحروف انكليزية كبيرة ولها معان خاصة بها تؤديها في برنامج باسكال ، ولا يجوز للمبرمج الذي يقوم بكتابة البرنامج بلغة الباسكال من إعادة تعريف هذه الكلمات أو استخدامها لغرض آخر غير المخصص لها ، ولهذا سميت كلمات محجوزة وهي كما يلي :

AND	FUNCTION	PACKED	VAR
ARRAY	GOTO	PROCEDURE	WHILE
BEGIN	IF	PRIGRAM	WITH
CASE	IN	RECORD	CONST
LABEL	REPEAT	DIV	MOD
SET	DO	MODULE	STRING
DOWNT0	NIL	THEN	ELSE
NOT	TO	END	OF
TYPE	FILE	OR	UNTIL
FOR	OTHERWISE	VALUE	

(3-2-9) مسميات المبرمج التعريفية (الأسماء التعريفية) Identifiers

وهو الاسم الرمزي للمتغير وهو أيضاً عبارة عن اسم لموضع تخزيني يعطى في البرنامج لتعريف قيمة الثابت أو المتغير أو الاجراء أو البرنامج , Constant , Variable , Procedure , Program ويستخدم لتعريف الاقترانات المكتبية ، ومسمى المبرمج التعريفية (Identifier) يتكون من حروف وأرقام في أية ترتيب كان باستثناء الرمز الأول من المسمى والذي يجب أن يكون حرفاً مع إمكانية استخدام الحروف الصغيرة أو الكبيرة ومن المفضل استخدام الحروف الصغيرة كي يتم تمييزها عن الكلمات المحجوزة التي تستخدم في اكثر الأحيان الحروف الكبيرة ويشترط في بعض الأحيان بان لا يزيد طول الاسم التعريفية عن ثمانية خانات بينما

يسمح للبعض الآخر بأية طول كان ... وفي لغة Turbo Pascal يستخدم الرمز (-) في كتابة الاسم شريطة ان لا يرد في بداية أو نهاية المسمى التعريفي للمبرمج. وفيما يلي بعض الأمثلة للمسميات التعريفية الصحيحة والمقبولة بلغة Turbo Pascal :

a	grade	tax	b6	m7
average	f16	product	sum	length
figure3	pressure	st-number	hours	rate
area	ss16			

أما الأسماء غير المقبولة في لغة Turbo Pascal فهي :

<u>الاسم التعريفي</u>	<u>سبب عدم القبول</u>
7-up	كونه يبدأ برقم وليس بحرف وكذلك لا يجوز استخدام الاشارة الخاصة (-)
3rd	كونه يبدأ برقم
B6.1	لكونه يستخدم الرمز الخاص (.)
X	لوجود رمز خاص فوق الحرف
VAR	لأنه كلمة محجوزة
B2	لاستخدامه حرف لاتيني

(4-2-9) الأسماء التعريفية القياسية Standard Identifiers

وفي بعض الأحيان تسمى المسميات التعريفية القياسية، وهي أسماء يشار إليها في كتابه اليرامج ويكون لها معنى معرف مسبقاً ومنها الإقترانات القياسية المثلثية وكذلك الإقترانات الأسية والإقترانات الرياضية المكتتبية والتي تسمى (Library Mathematical Functions) وكذلك الأسماء الدالة على نوع البيانات والإقترانات ويمكن إعادة تعريف هذه الأسماء القياسية على خلاف

الكلمات المحجوزة التي لا يمكن إعادة تعريفها ، وهذه الأسماء والاقتارات هي كما يلي :

ABS	IN	RESET
ARCTAN	MAXINT	REWRITELN
BOOLEAN	NEW	ROUND
CHAR	ODD	SIN
CHR	ORD	STRING
COS	OUTPUT	SQR
DISPOSE	PACK	SQRT
EOF	PAGE	SUCC
EOLN	PRED	TEXT
EXP	PUT	TRUE
FALSE	READ	TRUNC
GET	READLN	UNPACK
INPUT	REAL	WRITELN
INTEGER		

(9-2-5) الأعداد Numbers

الأعداد في Pascal هي عبارة عن سلسلة متصلة من الأرقام الحسابية والتي تبدأ بالعدد صفر 0 وتنتهي بالعدد 9 ويمكن للعدد أن يحمل الإشارة الموجبة أو السالبة ولا يمكن أن يحوي العدد على أية فارزة أو فراغ ، ويمكن للعدد أن يحوي الفاصلة العشرية (Decimal Point) أو أن تسبقه إشارة السالب (-) أو إشارة الموجب (+) والأعداد هي إما صحيحة أو حقيقية :

أ) الأعداد الصحيحة Integer Numbers :

العدد الصحيح في Pascal هو العدد الذي يكتب بدون فاصلة عشرية ويمكن أن يكون صفراً ، ويكتب دائماً بدون رمز الفاصلة العشرية . والأمثلة التالية توضح الأعداد الصحيحة :

0 47 +61 -1000 -20 31

اما الأعداد غير المقبولة كأعداد صحيحة في لغة Pascal فهي على سبيل المثال
كما يلي :

العدد	سبب عدم القبول
31.3	احتوائه على الفاصلة العشرية
612,12	احتوائه على الفارزة
#50	احتوائه على الرمز الخاص #
C16	احتوائه على الحرف الابداعي c
221.	احتوائه على الفاصلة العشرية
251 25	احتوائه على فراغ

ب) الأعداد الحقيقية Real Numbers

الأعداد الحقيقية هي الأعداد الموجبة أو السالبة أو الصفر والتي تكتب بوجود
الفاصلة العشرية ، ويمكن ان تكون هذه الأعداد في أحد الشكلين التاليين :

1- الشكل ذو الكسر العشري Decimal Notation

والأمثلة على العدد ذي الكسر العشري كما يلي :

50.16 -14.07 72.0 0.0 -0.617

ونشير هنا إلى أن العدد الحقيقي في الشكل ذي الكسر العشري يجب أن يحوي
رمز الفاصلة العشرية ، وبالإضافة إلى أنه عند كتابة العدد الحقيقي في الشكل ذي
الكسر العشري يجب وضع رقم عشري واحد على الأقل إلى يمين رمز الفاصلة
العشرية وكذلك رقم عشري واحد على الأقل إلى يسار رمز الفاصلة العشرية .

و الأرقام التالية غير مقبولة بلغة باسكال كأعداد حقيقية وللأسباب الموضحة كما
يلي :

العدد	سبب عدم القبول
4,561.1	لاحتوائه على الفارزة
6000	لا يحتوي على الفاصلة العشرية
D7.2	يحتوي على حرف الجدي
3 88.2	يحتوي على فراغ
.66	لا يحتوي على رقم إلى يسار الفاصلة
66.	لا يحتوي على رقم إلى يمين الفاصلة

2- الشكل المرفوع إلى أس Exponential Notation

وهي الأعداد المرفوعة إلى أس كما في الأمثلة التالية :

78.89E -36.8E6 16E5

ويستخدم هذا الشكل للأعداد المنتهية في الصغر أو المنتهية في الكبر فمثلاً يمكن

تمثيل العدد الحقيقي التالي بالشكل الأسّي وكما يلي :

0.000035 —————> 35.E-6
 —————> 35.0E-6
 —————> 3.5E-5
 —————> 0.35E-4

أو تمثيل العدد الحقيقي بالشكل التالي :

0.00234 —————> 2.34E-3

وهذا يعني ان العدد الموجود على يسار E مضروب بـ 10^{-3} ويمثل العدد الموجود إلى يمين E الأس الذي يرفع اليه العدد 10 ، مع ملاحظة أن الفاصلة العشرية المستخدمة في الشكل الأسّي يجب أن يكون على الأقل رقم إلى يمينها ورقم إلى يسارها ، كما أن الأس المستخدم إلى يمين E يجب أن يكون عدداً صحيحاً .

أما مدى الثوابت العددية فتختلف من جهاز حاسوب إلى آخر حسب نوع وطرز جهاز الحاسوب المستخدم ، ، أما عدد الأرقام المعنوية المسموح بها في العدد الحقيقي فهي تختلف حسب رقم الإصدار للغة Pascal وفي معظمها يسمح

بـ7 أو 8 أرقام . والجدول التالي يبين تمثيل بعض الأعداد بطريقة التدوين الأسّي بلغة باسكال :

القيمة	التدوين الرياضي	التدوين الاساسي
300000	$3.0 * 10^5$	3.0E5
300000	$3.0 * 10^5$	3.0E05
300000	$3.0 * 10^5$	+3.0E5
20000	$20.0 * 10^3$	20.0E3
0.763	$7.63 * 10^{-1}$	7.63E-1
0.000006	$6.0 * 10^{-6}$	6.0E-6

أما الامثلة التالية فهي غير مقبولة بالشكل الأسّي وذلك للأسباب الموضحة لكل منها كما يلي :

العدد	سبب عدم القبول
21.5E3.4	لان الأس ليس عددا صحيحاً
7.E-4	لعدم وجود رقم إلى يمين الفاصلة العشرية
.6E-8	لعدم وجود رقم إلى يسار الفاصلة العشرية
8.8 E 3	لوجود فراغ في اس العدد إلى يمين E

(9-2-6) الثوابت الرمزية String Constants

الثابت الرمزي في لغة Pascal هو عبارة عن سلسلة من الرموز المكونة من الحروف والأرقام والرموز الخاصة باستثناء رمز الحاصرة العلوية (أو تسمى في بعض الأحيان علامتي الاقتباس) يسمى عدد الخانات المحجوزة بين الحاصرات العلوية بطول اثبات الرمزي فمثلاً طول الثابت الرمزي "Ghazi Ibrahim" هو 13 رمزاً لأن الفراغ الموجود بين الاسمين يحسب رمزاً كونه احد رموز مجموعة الرموز الخاصة ، والأمثلة التالية تمثل ثوابت رمزية صحيحة .

- ‘ The speed of mind ‘
- ‘ I love PASCAL ‘
- ‘ Applied science University ‘
- ‘ P. O. Box 40 ‘

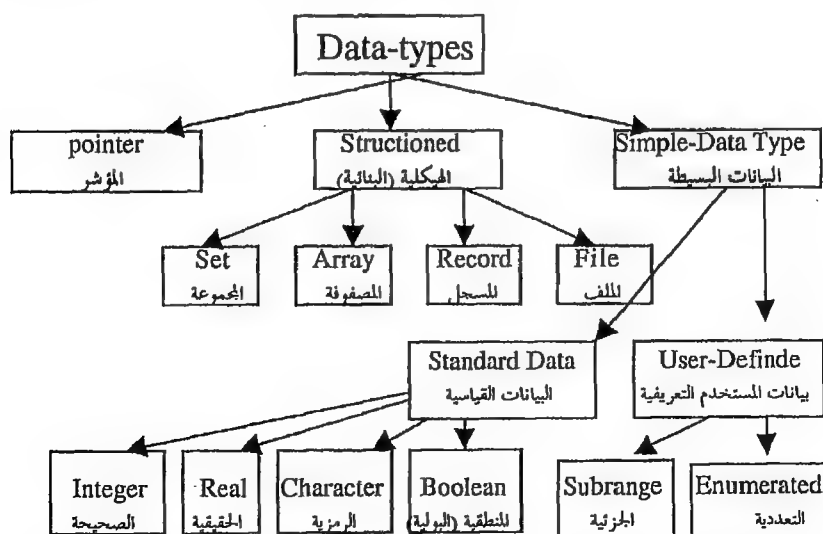
كل الثوابت الرمزية أعلاه وإن استخدمت أرقاماً حسابية داخلها (وذلك لأن هذه الأرقام لا تحمل أي قيمة أو معنى حسابي) تسمى بالثوابت الرمزية ، وتستخدم عادة كعناوين توضيحية مع النتائج الخاصة بالبرنامج .
ويختلف الحد الأقصى لعدد الخانات المسموح بها للثابت الرمزي من نوع إصدار باسكال إلى آخر ويسمح معظم أشكال باسكال بطول مقداره 255 رمزا .

(9-2-7) أنواع المعطيات (البيانات) Data Types

تتماز لغة Pascal عن لغات البرمجة الأخرى بقدرتها على التعامل مع أنواع مختلفة من المعطيات (البيانات) ، والشكل (1) يمثل انواع المعطيات .
وستعرض في هذا الفصل إلى المعطيات (البيانات) القياسية البسيطة والتي تعني بمتطلبات مقرر المقدمة ، ولزيد من المعلومات عن الأنواع الأخرى يمكن العودة إلى المراجع المتخصصة بالبرمجة الهيكلية بلغة باسكال .

أ) الثوابت Constants

وهي التي تحمل قيمة ثابتة لا تتغير أثناء تنفيذ البرنامج سواء أكانت هذه القيم قيم حسابية أو رمزية وبحيث يتم تعريف قائمة الثوابت في مقدمة برنامج Pascal



الشكل (1) انواع المعطيات (البيانات) التي يتعامل معها باسكال

وذلك قبل استعمالها فيه ، ويمكن توضيح الصيغة العامة لجملة تعريف الثوابت بلغة Pascal كما يلي :

```
CONST name1 = value1 ;  
      name2 = value2 ;  
      name3 = value3 ;  
      :  
      namen = valuen ;
```

CONST : حيث ان الكلمة المحجوزة تم التعبير عنها بـ

name : والاسم الثابت التعريفي بـ

value : وقيمة الثابت الحسابية أو الرمزية بـ

وفيما يلي بعض الأمثلة التوضيحية عن تمثيل الثوابت بلغة باسكال :

أ) الثابت الرمزي :

```
CONST address = ' science university
```

ب) الثابت الحسابي الحقيقي :

```
CONST distance = 16.4 ;
```

وسمي الثابت الحسابي حقيقي لان العدد في الطرف الأيمن في هذا المثال هو

عدد حقيقي لذلك يعتبر الثابت في الطرف الأيسر distance هو حقيقي أيضاً :

ج) الثابت الحسابي الصحيح :

```
CONST length = 105 ;
```

وسمي الثابت الحسابي صحيح لان العدد في الطرف الأيمن في هذا المثال هو عدد

صحيح ولذلك يعتبر الثابت في الطرف الأيسر Length صحيح ايضاً .

ومن أهم الثوابت العددية المحددة مسبقا في لغة باسكال هي كلمة MAXINT

والتي تمثل أكبر عدد صحيح يستوعبه الحاسوب .

Variables المتغيرات

المتغيرات هي التي تتغير قيمتها أثناء تنفيذ البرنامج ، والمتغير على عدة أنواع منها الحقيقي ومنها الصحيح ومنها الرمزي ومنها المنطقي ، لذلك يجب الإشارة إلى نوع المتغير في الجزء المخصص للتوضيح (أي الجزء الثاني في البرنامج الموضح في الفقرة 9-1) والصيغة العامة لجملة التعريف بنوع المتغير كما يلي :

```
VAR name1, name2, name3, .....namen : type;  
    name1, name2, name3, ..... namen : type;  
    .....  
    .....  
    name1, name2, name3,..... namen : type;
```

حيث : الاسماء التعريفية لاسماء المتغيرات (مسمى المبرمج التعريفي) هي :

name1 name2, name3,name_n

VAR : الكلمة المحجوزة لتعريف المتغيرات هي :

ملاحظة :

type : تعني نوع المعطيات أو البيانات ويمكن ان تكون احدى الأنواع

التالية :

REAL متغيرات حقيقية

INTEGER متغيرات صحيحة

BOOLEAN متغيرات منطقية

CHAR متغيرات رمزية

STRING متغيرات سلاسل زمنية

إذا كانت العملية هي باقي القسمة (Mod) فإن التعبير $A \bmod B$ تكون نتيجته 4 وجميع هذه النتائج هي نتائج صحيحة .

مثال (3): أوجد ناتج العمليات الصحيحة التالية .

العملية	النتيجة
$14+6$	20
$16-5$	9
$12*5$	60
$11 \text{ Div } 5$	2
$11 \text{ Mod } 5$	1

مثال (4): أوجد ناتج العمليات الصحيحة التالية :

العملية	النتيجة
$7 \text{ Div } 2$	3
$4 \text{ Div } 4$	1
$-7 \quad 2$	3
$-7 \text{ Div } 2$	-3
$7 \quad 2$	-3
$17 \text{ Mod } 10$	7

مثال (5): التعابير التالية غير مقبولة في لغة Pascal كتعابير صحيحة .

$19 \text{ Div } 14.0$

$7.5 \text{ Mod } 2$

ب) التعابير الحسابية الحقيقية :

يكون فيها حدي العملية الحسابية هو من النوع الحقيقي وتكون نتيجة

العمليات الحسابية قيمة عددية حقيقية و العمليات الحسابية المستخدمة هي

عمليات الجمع (+) ، والطرح (-) ، و الضرب (*) والقسمة (/)

مثال (6) : إذا كانت $A = 4.5$ ، $B = 5.3$ فإن :

$$A + B = 9.8$$

$$B - A = 0.8$$

$$B * A = 23.85$$

$$A / B = 0.85$$

ملاحظة : يمكن أن يكون أحد حدي العملية الحسابية صحيحاً والحد الآخر حقيقياً وفي هذه الحالة تكون نتيجة العملية الحسابية عدداً حقيقياً .

مثال (7) : إذا كانت $A = 11.2$ ، $B = 5$ فإن ناتج العمليات الحسابية الأربعة كما يلي :

$$A + B = 16.2$$

$$6.2$$

$$A * B = 56.0$$

$$A / B = 2.24$$

مثال (8) : التعابير التالية مقبولة في لغة Pascal على أساس أنها تعابير حسابية :

$$5 * 2 + 12 / 8$$

$$X * Y$$

$$X / Y$$

$$(a + b + c) / (a - b) + c$$

$$(x - y) + (x + y) / d$$

مثال (9) : التعابير التالية مقبولة في لغة Pascal على أساس أنها تعابير منطقية :

$$Y = X \quad \text{على أساس الجواب سيكون نعم أو كلا .}$$

$$200.0 > \text{Length} \quad \text{على أساس الجواب سيكون نعم أو كلا .}$$

$$\text{Factor} \leq 1.0 \quad \text{على أساس الجواب سيكون نعم أو كلا .}$$

ملاحظة : لا يمكن استخدام رمزين متتالين في تعبير واحد ، كذلك يجب معرفة أن لغة Pascal لا تستعمل الرفع إلى الأس بنفس الطريقة التي تستخدمها لغة

Basic أو Fortran وإنما تحول عملية الرفع إلى اس إلى عملية خطية باستخدام اللوغاريتم.

كذلك تستعمل لغة Pascal الاقترانات القياسية (المكتبة)

Standard (Library) Functions

وهذه الاقترانات تكون جاهزة في الحاسوب للاستعمال الفوري ، وتسمى في لغات البرمجة الأخرى الاقترانات المبنية داخلياً (وعلى سبيل المثال يعتبر اقتران $\text{Cos}(x)$ اقتران قياسي يمكن إيجاد قيمته حال إعطاء قيمة (x) مع العلم انه يمكن أن تكون قيمة (x) بيان عددي أو منطقي أو رمزي أو حتى تعبير حسابي أو منطقي .

وتقسم الاقترانات القياسية إلى أربعة مجموعات :

- (1) اقترانات قياسية عددية نتيجتها أعداد صحيحة .
 - (2) اقترانات قياسية عددية نتيجتها أعداد حقيقية .
 - (3) اقترانات منطقية نتيجتها إما صح أو خطأ .
 - (4) اقترانات رمزية تكون نتيجتها مستندة إلى النظام المستخدم في التشفير .
- ولو عدنا الآن إلى كل مجموعة من هذه الاقترانات لوجدنا ما يلي :
- أولاً : ان الاقترانات القياسية العددية والتي تكون نتيجتها أعدادا صحيحة هي كما يلي :

- اقتران لحساب تربيع الأعداد الصحيحة $\text{SQR}(x)$
- اقتران لإيجاد القيمة المطلقة للعدد الصحيح $\text{ABS}(x)$
- اقترانات حذف الجزء الكسري للعدد الحقيقي $\text{TRUNC}(x)$
- اقترانات الجزء الصحيح في العدد الحقيقي $\text{INT}(x)$
- اقتران تقريب العدد الحقيقي إلى اقرب عدد صحيح $\text{ROUND}(x)$

مثال (10) :

$$\text{ABS}(5) = 5$$

$$\text{ABS}(-5) = 5$$

$$\text{ROUND}(4.5) = 4$$

$$\text{ROUND}(4.6) = 5$$

$$\text{SQR}(3) = 9$$

$$\text{SQR}(-3) = 9$$

$$\text{TRUNC}(4.2) = 4$$

$$\text{TRUNC}(4.6) = 4$$

$$\text{INT}(5.6) = 5$$

ثانياً : أما الاقترانات القياسية التي تكون نتيجة تنفيذها أعداداً حقيقية فهي جميع

الاقترانات مثل :

- اقتران الجذر التربيعي $\text{SQR}(x)$

- اقتران القيمة المطلقة $\text{ABC}(x)$

- اقتران اللوغاريتم الطبيعي $\text{LN}(x)$

- اقتران تربيع العدد $\text{SQR}(x)$

- اقتران الظل المعكوس $\text{ARCTAN}(x)$

- الأقران الأسّي لحساب e^x $\text{EXP}(X)$.

مثال (11) :

$$\text{ABS}(3.2) = 3.2$$

$$\text{ABS}(-3.2) = 3.2$$

$$\text{SQR}(2.5) = 6.25$$

$$\text{SQR}(-2.5) = 6.25$$

$$\text{SQRT}(5) = 2.23$$

$$\text{SIN}(9) = 0.156$$

$$\text{COS}(-1.2) = 0.9998$$

$$\text{LN}(2) = 0.693$$

$$\text{EXP}(3) = 20.085$$

$$\text{EXP}(-3) = 0.0497$$

ثالثاً : أما الاقتترانات المنطقية فهي التي يكون نتيجة تنفيذها إما True أو False وصيغتها العامة :

$$G_1 \quad R \quad G_2$$

حيث G_1 : يمكن ان يكون تعبير حسابي أو ثابت منطقي أو رمزي .
 R : معامل منطقي ويمكن أن يكون أصغر ($<$) أو أصغر أو يساوي (\leq) أو يساوي ($=$) أو أكبر ($>$) أو أكبر أو يساوي (\geq) أو لا يساوي (\neq) .
 كلمات NOT/AND/OR ومهما كان التعبير المنطقي بسيطاً أو مركباً فان قيمتها دائماً هي إما True أو False .

مثال (12) : إذا كانت قيمة a صح True

وقيمة b خطأ False

وقيمة c صح True

فإن قيمة التعبير المنطقي التالي صح True :

a AND (b OR c)

وقيمة التعبير المنطقي التالي خطأ False :

(a AND b) OR (b AND c)

وقيمة التعبير المنطقي التالي خطأ False :

a AND NOT b OR NOT c

رابعاً : أما الاقتترانات الرمزية فهي الاقتترانات التي تعتمد على وضع حاصرتين علويتين ولهما علاقة بنظم الشيفرة لتبادل المعلومات مثل ASCII المستخدمة مع معظم اجهزة الحواسيب الشخصية (أنظر الجدول (1) الفصل الرابع) وهذه الاقتترانات هي :

- اقتران $PRED(x)$ وهو اقتران تحديد الرمز الذي يسبق الرمز (x) في

مجموعة الرموز ويسمى Predecessor function

- اقتران $SUCC(x)$ وهو اقتران تحديد الرمز الذي يلي الرمز (x) في مجموعة الرموز ويسمى **Successor Function** .

- اقتران $ORD(x)$ وهو اقتران إيجاد القيمة الداخلية للرمز (x) ويسمى **Ordinary Function** .

- اقتران $CHR(x)$ وهو اقتران إيجاد الرمز الخارجي الذي يقابل العدد الصحيح (x) ويسمى **Character Function**

(9-2-9) قاعدة الأولوية Rule of Precedence

قاعدة الأولوية هي القاعدة المستخدمة في معالجة العمليات الحسابية وفق تسلسل خاص بحيث يتم معالجة البيانات وبالتسلسل التالي :

1. معالجة ما داخل الأقواس .

2. معالجة عملية الضرب (*) وعملية القسمة (DIV) وعملية الباقي من القسمة (MOD) وعملية AND وعملية (/) وهذه العمليات لها نفس الأولوية .

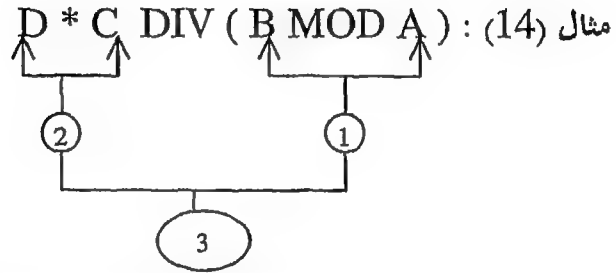
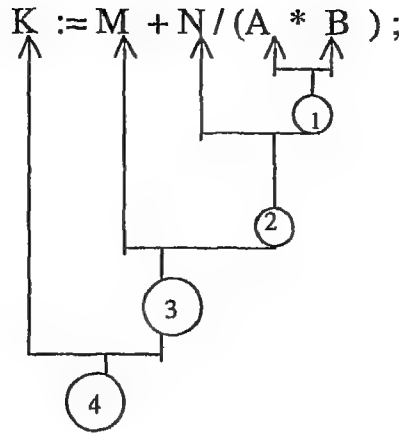
3. معالجة عملية الجمع (+) والطرح (-) وعملية OR ، ولهما نفس الأولوية .

مع الأخذ بعين الاعتبار ما يلي :

أ- العمليات الحسابية التي لها نفس الأولوية تنفذ على الترتيب حسب ظهورها من اليسار إلى اليمين .

ب - اذا كانت الأقواس متداخلة تعطي الأولوية في التنفيذ إلى القوس الداخلي ثم الخارجي الذي يليه ثم الخارجي الذي يليه، أي أن نبدأ من الداخل الى الخارج .

مثال (13) : لاستخدام قاعدة الأولوية في معالجة المعادلة التالية تصبح كالآتي :



(9-2-10) جمل الإسناد Assignment Statement

- جملة الإسناد (جملة التعيين) تستعمل لتخصيص قيمة محددة إلى متغير ،

والصيغة العامة لجملة الاسناد في لغة باسكال هي :

Variable := Expression

حيث ان :

Variable: هو متغير قد يكون من أي نوع .

Expression : هو تعبير وقد يكون ثانيا (عدديا أو منطقيا) أو متغيرا

من نفس نوع المتغير الموجود في الطرف الأيسر .

مثال (15) : الجملة التالية هي جملة إسناد حسابية

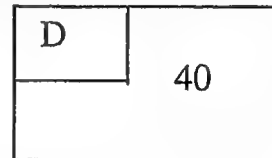
$$\text{Zakat} := 0.025 * \text{Total Money} ;$$

حيث الفارزة ؛ المنقوطة تعني فصل هذا السطر عن السطر اللاحق فلو فرضنا ان Total Money كانت قيمته مساوية إلى 1000 مثلا فإن النتيجة 25 سوف يتم خزنها في مخزن المتغير الخاص بـ Zakat

مثال (16) : جملة الاسناد التالية :

$$D := 40 ;$$

تعني اسناد القيمة العددية 40 للمتغير العددي D



مثال (17) : جملة الإسناد الحسابية التالية :

$$\text{Number} := \text{Number} + 1 ;$$

تعني إضافة القيمة 1 إلى قيمة المتغير Number وإسناد النتيجة ثانية للمتغير Number وهذا يعني أن جملة الإسناد تصبح فيها قيمة المتغير Number الجديدة تساوي قيمة Number مضافا إليها 1 .

(3-9) المكونات الهيكلية لبرنامج باسكال:

Structure Of Pascal Program

كما أوضحنا سابقا فان أي برنامج بلغة باسكال يتكون من ثلاث مقاطع رئيسيه :

المقطع الأول : وهذا المقطع مخصص لعنوان البرنامج أو ما يسمى :

The Program Heading

المقطع هذا يتكون في كثير من الأحيان من سطر واحد على شرط أن يبدأ بكلمة

محموزة وهي كلمة PROGRAM ثم يليها اسم البرنامج (program-name) مع ملاحظة ان هذا المقطع يجب أن يخضع في كتابته إلى قواعد كتابة المسميات التعريفية ، ثم يتبع اسم البرنامج وعنوانه اسلوب أو نوع ملف الإدخال والإخراج ويمكن في بعض الحالات عدم تحديد اسلوب ملف الإدخال في حالة كون الاسلوب معروف من قبل الحاسوب والصيغة العامة للمقطع الأول كما يلي:

ROGRAM x-name (INPUT,OUTPUT,FILE-NAME,..)

وفي بعض الأحيان يمكن اضافة سطر توثيقي محصور بين حاصرتين وقوسين للتعريف باسم ونوع وعمل البرنامج وهذا السطر يكون فقط للشرح .

ونشير هنا إلى أننا نكتب كلمة الادخال INPUT في عنوان البرنامج في حالة كون تنفيذ البرنامج يتطلب قراءة بيانات من لوح المفاتيح (Keyboard) وكلمة (OUTPUT) في حالة كون تنفيذ البرنامج يتطلب إظهار النتائج على وسط الإخراج من خلال جهاز الإخراج ، علماً بأن هذين الملفين ملف الإدخال INPUT وملف الإخراج OUTPUT هما قياسيان ولا يجوز تعريفهم تأتي في بداية السطر . أما فهو يمثل اسم الملف الخارجي والذي يتم تعريفه ضمن البرنامج مع ملاحظة أنه في حالة وجود أكثر من ملف وأكثر من اسم واحد للملف فيجب كتابته جميع أسماء الملفات و يجب أن نضع الفاصلة (,) بين كل اسم ملف وآخر .

المقطع الثاني : يسمى هذا المقطع بمقطع التوضيحات أو التعريفات أو التصريحات وهي (Declartion Section) ويتكون هذا المقطع من جملة واحدة أو أكثر من جملة وذلك للتعريف أو للاعلان عن العناوين ، أو البيانات ، أو الثوابت ، أو الإجراءات، أو الاقترانات المستخدمة في البرنامج مع ضرورة معرفة التابع المنطقي للتعريفات حسب التسلسل التالي :

LABEL
CONST
TYPE
VAR
PROCEDURE
FUNCTION

حيث أن هذا التعريفات يجب أن تكون متسلسلة كما في أعلاه في حالة احتواء البرنامج الواحد على أكثر من تصريح أو توضيح .

المقطع الثالث : وهو المقطع التنفيذي (Executable Section) أو ما يسمى في بعض الأحيان Bluk ، ويضم تعليمات وجمل البرنامج المطلوب تنفيذها لتحقيق الهدف المراد من تصميم البرنامج وتنفيذه ، ويكون هذا المقطع دائما محصور بين الكلمتين المحجوزتين BEGIN و END ، والصيغة العامة كما يلي:

BEGIN;;; END.	} جملة وتعليمات تنفيذية للبرنامج
---	-------------------------------------

حيث أن كلمة BEGIN تمثل بداية البرنامج ولا يتبعها أية اشارة وتمثل بداية البرنامج التنفيذي بينما ينتهي البرنامج بكلمة END يتبعها نقطة (.) إلزامية وهي تمثل نهاية البرنامج .

أما الجمل والتعليمات التي تأتي داخل الكلمتين BEGIN و END ، فيجب ان تنتهي بفاصلة منقوطة (;) Separator ، وهنا نشير إلى ان القسم التنفيذي يمكن ان يضم اكثر من BEGIN و END كمقاطع أو أجزاء داخلية ، وبالتالي يجب ان تنتهي كل END بفاصلة منقوطة ما عدا END الاخيرة التي تمثل نهاية البرنامج فتنتهي بـ نقطة .

مثال (18) : المثال التالي يوضح برنامج بلغة باسكال لاجاد مساحة ومحيط دائرة نصف قطرها معلوما .

الخطوة الأولى : قانون مساحة الدائرة هو :

المساحة = نصف القطر * نصف القطر * النسبة الثابتة

$$\text{Area} := \text{Pi} * \text{SQR}(\text{Radius})$$

باعتبار ان $\text{Pi} = 3.14159$ وهي النسبة الثابتة

باعتبار ان (Radius) هو نصف القطر

قانون محيط الدائرة هو :

المحيط = النسبة الثابتة * نصف القطر * 2

$$\text{Circumference} := 2 * \text{Pi} * \text{Radius} ;$$

باعتبار ان Circumference هو محيط الدائرة .

اذا هذه المسألة احتوت على ثوابت وهي النسبة الثابتة : $\text{Pi}=3.14159$

والمتغيرات هي نصف قطر الدائرة المعلوم Radius كمدخلات ، ومساحة الدائرة

Area ومحيط الدائرة Circumference كمخرجات ويمكن ان يكون نصف

قطر الدائرة كمخرجات بالإضافة إلى كونه كمدخلات ، والعلاقات هي القوانين

الخاصة بالمساحة والمحيط للدائرة .

الخطوة الثانية : تحديد ما تم شرحه في الخطوة الأولى وبالشكل التالي :

الثوابت : $\text{Pi}=3.14154$

المتغيرات :

المدخلات : Radius : نصف القطر

المخرجات : Area : مساحة الدائرة

Circumference : محيط الدائرة

Radius : نصف القطر

إذاً البرنامج يكون على الشكل التالي حسب المقاطع الثلاث التي ذكرناها سابقا والتي يتكون منها برنامج Pascal :

المقطع الأول	{	PROGRAM GHAZI (Input , Output) ; (*PROGRAM to compute and print the Area and Circumference of a given circle*)
		سطر تعريف غير ملزم وغير منفذ
المقطع الثاني	{	CONST
التعريفات أو		Pi = 3.14 159 ;
التصريحات	{	VAR
	{	Radius ,AREA ,Circumference : REAL ;
المقطع		BEGIN
الثالث		WRITELN ('ENTER The Value of Radius') ;
المقطع		READ (Radous) ;
التنفيذي		AREA := Pi * SQR(Radius) ;
		Circumference := z* Pi * Radius ;
		WRITELN('FOR Radius = ' , Radius) ;
	WRITELN('The Area = ' , Area) ;	
	WRITELN('Circumference=' , circumferenc);	
	{	END .

ثم بعد كتابة وادخال البرنامج إلى الحاسوب يتم اعطاء قيمة نصف القطر Radius ثم يتم الضغط على مفتاح الإدخال ENTER وتظهر النتائج على الشاشة .

(4 - 9) أساليب إدخال وإخراج البيانات في برنامج باسكال

ستعرض في هذه الفقرة إلى الأساليب المتبعة في نقل وإدخال البيانات ذات القيم الحسابية أو الرمزية وكذلك سوف نتعرض إلى أساليب إخراج وطباعة نتائج المدخلات بعد إجراء المعالجات عليها .

Input Statements (9-4-1) جمل الإدخال

يتم إدخال البيانات إما عن طريق ادخال قيم البيانات مع البرنامج نفسه باستخدام جملة الاسناد Assignment Statement وتسمى أيضا جملة التعيين ، أو عن طريق تسجيل البيانات في ملف يدعى ملف المدخلات حيث يتم قراءة قيم المتغيرات منه باستعمال جمل القراءة أثناء تنفيذ البرنامج أو بالإدخال المباشر باستخدام جمل القراءة أيضا .

أولا : استخدام جملة الإسناد

كما وضحنا في الفقرة (9-2-10) فإن جملة الإسناد تأخذ الشكل التالي :

VARIABLE := EXPRESSION

وهذا يعني أنه يتم إسناد قيمة التعبير Expression الموجود في الطرف الأيمن إلى المتغير الموجود في الطرف الأيسر ، وتخزن قيمة التعبير باسم المتغير ، ويتم التعامل في البرنامج أينما وجد المتغير مع قيمته المخزنة . وعادة يستخدم هذا الأسلوب عندما تكون عدد المتغيرات قليلا والبرنامج صغيرا .

مثال (19) :

Sum := 5032.5

حيث يتم تخزين العدد 5032.5 باسم المتغير Sum ، وبالتالي أينما وجدت كلمة Sum في البرنامج يتم تبديلها بقيمتها مباشرة . المؤشرة أعلاه

مثال (20) : إذا كان :

S := 50;

K := 5;

عندئذ عند ظهور الجملة التالية في البرنامج .

mean := S/N;

يتم تبديل S بـ 50 و N بـ 5 وتجري عملية القسمة ويخزن الناتج الذي يساوي 10 في المتغير mean .

ثانيا : استخدام ملف المدخلات :

يستخدم هذا الأسلوب عندما تكون عدد المتغيرات في البرنامج كثيرة وحجم البيانات كبيرا ، حيث تقرأ البيانات الموجودة على ملف الإدخال Input File بصورة آلية من خلال الجمل والأوامر الخاصة بقراءة الملفات ، وأهم ميزات هذا الأسلوب هو إمكانية تنفيذ البرنامج عددا من المرات يساوي إلى عدد السجلات الموجودة في الملف حيث يقوم البرنامج بتكرار الخوارزمية المطلوبة من أجل جمع بيانات الملف .

ثالثا : التخاطب المباشر مع الحاسوب :

وذلك باستخدام جملة القراءة READ , READLN

أ. جملة القراءة (الإدخال) READ Statement :

هذه الجملة تستخدم في لغة Pascal لقراءة وإدخال قيم المتغيرات الحسابية أو الرمزية إلى الحاسوب لاستخدامها في البرنامج وصيغتها العامة كمايلي:

READ (Variable ₁ , Variable ₂ Variable _n)

حيث نجد أن جملة READ تمثل مسمى تعريفي قياسي ويعامل معاملة أحد الكلمات المحجوزة أما (Variable) فهي إحدى مسميات المبرمج لاسماء المتغيرات وقد تكون متغيرات عديدة أو رمزية .

كما نلاحظ في الصيغة العامة اعلاه ان اسم المتغير المراد ادخال قيمته الحسابية أو الرمزية تكون موضوعة ومحصورة بين قوسين وتسبقها جملة READ مع ملاحظة هامة ، وهي اننا قمنا باستثناء المتغير المنطقي وذلك لعدم امكانية ادخال المتغير المنطقي باستعمال جملة القراءة READ .

ويتم تنفيذ أمر قراءة المتغيرات بالترتيب الذي وضعت عليه وحسب الوصف المعرفة فيه وبشكل تسلسلي ومن خلال إدخال هذه القيم من لوحة المفاتيح مع مراعات بعض الملاحظات التي سنوضحها من خلال الأمثلة التالية :

مثال (21): ليكن لدينا البرنامج التالي الذي يضم بعض جمل القراءة .

```
PROGRAM Exam (Input,Output);
VAR
    F,C,N,S,L: INTEGER;
BEGIN
    READ (N,C,F);
    S:=F + C;
    L:=N +F;
    Writeln ('S=',S,'L=',L);
END.
```

ولدى تنفيذ البرنامج تظهر شاشة المخرجات (الشاشة السوداء) ويظهر المؤشر (-) cursor في الجهة اليسرى من الشاشة ، ويجب على المستخدم إدخال قيم المتغيرات من لوحة المفاتيح بنفس التسلسل التي وضعت فيه في جملة القراءة إما على سطر واحد على أن يفصل بينها مسافة لا تقل عن فراغ واحد ثم ضغط مفتاح الإدخال Enter ليتم تخزين تلك القيم المدخلة باسم المتغيرات ثم متابعة تنفيذ البرنامج ، أو إدخال كل قيمة على سطر يتبعها ضغط مفتاح الإدخال Enter . أي ان شاشة الإخراج عند تنفيذ البرنامج ستكون على الشكل التالي :

4	5	6	↓
S=11		L=10	

أو

4	↓
5	↓
6	↓
S=11	L=10

حيث اشارة ↓ تعني أنه يجب ضغط مفتاح الإدخال Enter .

ملاحظة : إن المتغيرات في جملة القراءة في المثال (21) السابق يمكن كتابتها

ضمن أكثر من جملة قراءة وعلى الشكل التالي :

READ (N,C);
READ (F);

}

READ (N);
READ (C,F);

}

أو

READ (N);
READ (C);
READ (F);

}

أو

وعند التنفيذ يتم إدخال قيم المتغيرات كما ذكرنا سابقا إما على سطر واحد يفصل بينها فراغ واحد على الأقل ثم مفتاح ENTER أو كل قيمة على سطر يتبعها . ENTER

ب. جملة القراءة (الإدخال) READLN Statement

لا تختلف هذه الجملة عن جملة الإدخال READ في وظيفتها وإنما تختلف من حيث الأداء ، حيث أن READLN تعني القراءة على نفس السطر الموجود عليه المؤشر والصيغة العامة لها نفس الصيغة للجملة READ .

مثال (22) : إذا كتبنا جملة القراءة في البرنامج السابق المثال (21) بالشكل التالي:

READLN (N,C,F);

فإنه يتم الإدخال بنفس الأسلوب الموضح في المثال (21) ، أما إذا كتبنا جملة القراءة بالشكل التالي :

READLN(N,C);
READ (F);

عندئذ يجب إدخال قيمة N,C على السطر الأول بحيث يفصل بينها فراغ ثم الضغط على مفتاح ENTER ثم قيمة F على السطر التالي ، وإذا حدث وأن كتبنا القيم الثلاث على سطر واحد فإنه لا يأخذ حسب الجملة الأولى إلا قيمتين فقط ويطلب من المستخدم إدخال القيمة الثالثة على سطر جديد.

هذا يعني إن وجود LN في نهاية الجملة READ تعلم الحاسوب بنقل ميكانيكية قبول المدخلات من السطر التالي الذي يليه .

ملاحظة : عند كتابة جملة الإدخال بالشكل التالي :

READLN;

في نهاية البرنامج وقبل جملة END الأخيرة ، تعطي إمكانية رؤية نتيجة تنفيذ البرنامج على شاشة المخرجات ، على أن يسبقها جملة READLN في البرنامج بأي شكل ، وبالضغط على مفتاح Enter نعود إلى شاشة باسكال المكتوب فيها البرنامج .

(2-4-9) **OUTPUT Statement** جمل الإخراج

يتم إخراج نتائج تنفيذ البرنامج إما على الشاشة مباشرة أو على الطابعة أو تخزين النتائج على ملف يدعى ملف الإخراج Output File، وفي جميع الحالات نستخدم الجملة WRITE أو WRITELN مع اختلافات بسيطة حسب نوع الإخراج .

WRITE Statement جملة الإخراج غير الموصوفة

تستخدم هذه الجملة لإخراج النتائج على الشاشة والصيغة العامة لها كما

يلي :

WRITE (List) ;

حيث أن list تمثل ثوابت أو متغيرات Variables أو تعابير حسابية أو منطقية Expression ، كما يمكن أن تحتوي على جمل و تعليقات لشرح نوعية المخرجات على أن توضح هذه التعليقات ضمن حاصرتين علويتين وليس بالضرورة أن تضم list نوع واحد كما ذكر آنفا بل يمكن أن تكون خليط من جميع الأنواع وهذا يتعلق بالمرمج وشكل النتائج المراد إخراجها ، ويفضل دوما كتابة

تعليقات قبل كل نوع ثمن ماهية النتائج وخاصة عند التعامل مع برامج تضم حجم كبير من النتائج .

ونشير هنا إلى أن قائمة list من الأنواع السابقة يجب أن تفصل بينها الفارزة (commans) .

مثال (23) : إذا كانت قيم المتغيرات التالية في البرنامج كما يلي :
A = 10 , B = 20 , C = 30 , D = 40 , E = 50 ,
التالية في البرنامج

WRITE (A,B,C,D,E) ;

تظهر النتائج على شاشة المخرجات كما يلي على سطر واحد

1020304050

نلاحظ أن النتائج لا تفصل بينها أية فراغات ، لذلك يجب وضع فراغات بينها عن طريق ترك فراغ أو أكثر ضمن حاصرتين ضمن جملة الطباعة بالشكل التالي :

WRITE (A,' ' ,B,' ' ,C,' ' ,D,' ' ,E);

10 20 30 40 50

مثال (24) لطباعة قيم تعبير حسابي باستخدام جملة الطباعة WRITE بفرض أن قيمة a=7 , b=5 نستخدم جملة الطباعة التالية :

WRITE ('D=' , a-b) ;

فإن شكل الطباعة كنتيجة سوف تظهر كما يلي :

d = 2

مثال (25) : لطباعة قيم لثوابت رمزية باستخدام جملة الطباعة WRITE على الشكل التالي :

WRITE ('GHAZI,' , MOUHAMAD',' , IBRAHEM')

فإن النتيجة ستكون بالشكل التالي :

GHAZI MOUHAMAD IBRAHEM

مع ملاحظة الفراغات فكلما زادت الفراغات داخل الحواصر العلوية لكل ثابت رمزي كلما زادت الفراغات في الطباعة واطهار .

مثال (26): في بعض الأحيان يطلب اظهار تعليقات على المتغيرات مع قيمتها عند

الطباعة فلو فرضنا ان قيم $A=15.2$, $B=62.2$, $C=14$, $D=20$, $E=6$

وكانت جملة الطباعة WRITE بالشكل التالية :

```
WRITE ('A=',A,' B=',B,' C=',C);
```

```
WRITE (' D=',D,' E=',E);
```

عندئذ تظهر النتائج على شاشة المخرجات كما يلي :

```
A =15.2 B =62.2 C =14 D =20 E =6
```

(ب) جملة الإخراج غير الموصوفة **WRITELN Statement**

إن جملة الطباعة **WRITELN** والتي تعني الطباعة بسطر جديد لا تختلف كثيرا عن جملة الطباعة غير الموصوفة **WRITE** في وظيفتها اذ ان الاختلاف هو في طريقة الأداء في الطباعة حيث ان الصيغة العامة لهذه الجملة هي كما يلي :

```
WRITELN (List) ;
```

حيث **List** تمثل قوائم الطباعة والإخراج والتي من الممكن أن تكون واحد او اكثر من الأنواع التالية :

1. Variables متغيرات حقيقية أو صحيحة أو رمزية
2. Constatants ثوابت
3. Strings ثوابت رمزية
4. Expressions تعابير حسابية أو منطقية

على أن يفصل بين هذه المكونات الفارزة (,) (Commons) .

إن وجود LN في نهاية الجملة WRITE يعطي الأمر إلى الحاسوب بنقل ميكانيكية الطباعة إلى السطر التالي .

ملاحظة (1) : ان استخدم امر الطباعة السابقة بالشكل التالي :

WRITELN;

يؤدي إلى ترك سطر فارغ ونقل ميكانيكية الطباعة إلى السطر التالي .

مثال (27) : اذا كانت قيم :

. A= 4, B= 6, C= 12.5, D= 12, E= 14, F= 5.2

فعند استخدام جملة WRITELN التالية :

WRITELN (A,B,C,D,E,F) ;

فإن النتيجة ستظهر كما يلي :

4612.512145.2

وعند استخدام جملة WRITELN بالشكل التالي :

WRITELN (A) ;

WRITELN (B);

WRITELN (C,D);

WRITELN (E,' ',F) ;

فإن النتيجة ستظهر كما يلي :

4
6
12.512
14 5.2

ج) جمل الطباعة الموصوفة Formatted Writeln

لتوصيف البيانات المخرجة والتحكم بشكل الإخراج نستخدم جملة

WRITE ، WRITELN بنفس الأسلوب السابق مع بعض الإضافات حسب

نوع البيانات .

لطباعة العناوين والثوابت الرمزية ، نستخدم الصيغة التالية :

```
WRITE ('String Constant':N,..... );
```

حيث : String Constant ثوابت رمزية .

N طول الحقل المخصص للطباعة .

ويمكن استخدام WRITELN أيضاً بنفس الصيغة

مثال (28) :

WRITELN ('1245 AY');

```
WRITELN ('PASCAL':10, ' ':5, 'Programming':10);
```

وعند التنفيذ تكون شكل المخرجات كما يلي :

[illegible]

من خلال المثال السابق نلاحظ ان الجملة الاولى هي بدون توصيف وظهرت النتائج ابتداءً من العمود الأول ، بينما في الجملة الثانية عندما وصفنا الثابت الرمزي الاول ب 10 عندئذ يقوم الحاسوب بحجز العشرة أعمدة الأولى لوضع كلمة Pascal ويكتبها في أقصى اليمين من الحقل المخصص ، ثم يحجز خمسة أعمدة ويتركها فارغة ، ثم يقوم بحجز عشرة أعمدة أخرى لكلمة Programming ، إلا ان الكلمة الأخيرة نحتاج الى اكثر من عشرة اعمدة عندئذ يهمل الحاسوب طول الحقل المخصص ويكتبها ضمن (11) عامود .

لطباعة الأعداد الصحيحة ، نستخدم الصيغة التالية :

```
WRITE (Variable:N,..... );
```

حيث Variable يمثل اسم المتغير المعروف على انه صحيح Integer

N هو طول الحقل المخصص لطباعة العدد الصحيح. بما فيه الاشارة

مثال (29) : اذا كان :

$$A=8451, \quad B=-89, \quad C=12354$$

فإن أمر الطباعة التالي :

WRITELN (A:6,B:4,' ' :3,C:3);

يظهر عند التنفيذ النتائج بالشكل التالي :

		8	4	5	1		-	8	9				1	2	3	5	4			...
--	--	---	---	---	---	--	---	---	---	--	--	--	---	---	---	---	---	--	--	-----

من هذا المثال نلاحظ ان الحاسوب حجز 6 أعمدة الأولى لوضع قيمة A ويكتبها في أقصى اليمين من الحقل المخصص لـ A ، ثم يحجز الاعمدة الثلاثة التالية ويتركها فارغة ، ثم يحجز الاعمدة الثلاثة التالية لوضع قيمة C والتي تحتاج الى خمسة أعمدة عندئذ الحاسوب يتجاهل التوصيف المحدد بجمله الطباعة ويكتب قيمة C ابتداءً من العمود الذي يلي الحقل المخصص للفراغات الثلاث .

ولطباعة الأعداد الحقيقية ، تستخدم الصيغة التالية :

WRITE (Variable: N:M,...);

حيث : Variable هو اسم المتغير المعروف على أنه حقيقي Real .

N طول الحقل المخصص لطباعة العدد الحقيقي بجزئية الصحيح

والكسري والفاصلة العشرية والإشارة .

M طول الحقل المخصص لطباعة الجزء الكسري فقط .

مثال (30) : اذا كان :

$$X=28.738, \quad Y=132.4, \quad Z=5.6$$

فإن أمر الطباعة التالي :

WRITELN (X:6:2,' ' , Y:4:2,' ' :3,Z:4:2);

		2	8	.	7	4					1	2	3	.	4	0			5	.	6	0	..
--	--	---	---	---	---	---	--	--	--	--	---	---	---	---	---	---	--	--	---	---	---	---	----

من المثال نلاحظ ان الحاسوب من خلال التوصيف سيقوم بحجز الأعمدة الستة الأولى لوضع قيمة X ، ويكتبها في أقصى يمين الحقل المخصص .

مع الأخذ بعين الاعتبار ان القسم الكسري مخصص له فقط خانتين أي عمودين لذلك يقوم الحاسوب تلقائيا بتدوير الجزء الكسري الى خانتين فقط ، يلي الأعمدة الستة الأولى أربعة أعمدة فارغة ثم يليها أربعة أعمدة لقيمة Y ، الا ان قيمة Y تحتاج الى اكثر من أربعة أعمدة لذلك يتجاهل الحاسوب التوصيف ، ويقوم فقط بتوصيف الجزء الكسري أي انه يكتب الجزء الصحيح بأكمله وفي الجزء الكسري خانتين ، بمعنى آخر ان قيمة Y احتاجت الى 6 أعمدة ، بعد قيمة Y مباشرة بترك ثلاثة أعمدة فارغة ثم يحجز بعدها أربعة أعمدة لوضع قيمة Z حيث يكون مخصص فقط عمودين للجزء الكسري .

ملاحظات حول قواعد التوصيف :

- 1- يتم طباعة البيانات في أقصى يمين الحقل المخصص في التوصيف .
- 2- اذا كان عدد رموز الثابت الرمزي او عدد خانات العدد المراد طباعتها اكبر من عدد الأعمدة المخصصة له ، عندئذ يتجاهل الحاسوب التوصيف ويطبعها وكأنه لا يوجد توصيف أصلا .
- 3- إذا كان عدد خانات الجزء الكسري في العدد الحقيقي المراد طباعته اكبر من عدد الأعمدة المحجوزة له في التوصيف عندئذ يقوم الحاسوب بتدوير الجزء الكسري حسب الطلب .
- 4- اذا كان عدد خانات الجزء الكسري في العمود الحقيقي المراد طباعته اقل من عدد الأعمدة المحجوزة له في التوصيف عندئذ تملئ الأعمدة الزائدة بأصفار .

Page فتح صفحة جديدة

باستخدام جملة الاخراج او الطباعة WRITELN وفي حالة الحاجة

لطباعة الجملة في صفحة جديدة يمكن ان نستخدم الجملة التالية للطباعة :

WRITELN (C,D,E);

PAGE;

WRITELN (E,F,G);

في هذا المثال نلاحظ انه سيتم طباعة C,D,E في سطر فيما يتم طباعة C,D,E في سطر جديد وصفحة جديدة ، علماً أن إصدارات باسكال ليست جميعها تقبل صيغة الصفحة الجديدة هذه .

مثال (31): افترض أن قيم المتغيرات هي كما موضح صورتها أدناه وضح كيف ستكون صورة طباعتها باستخدام جملة الطباعة الموضحة .

المتغير	نوع المتغير	قيمة المتغير
E	صحيح	-80
F	حقيقي	11.2
G	رمزي	GHAZI
H	منطقي	FALSE

باستخدام جمل الطباعة التالية :

WRITELN ('E=' , E:4, ' ':3);

WRITELN ('F=' , F);

WRITE ('G=' , G);

WRITELN ('H=' , H);

ستكون النتيجة كما يلي :

E =	-	80				F =	11.2				
G =	G	H	A	Z	I	H =	F	A	L	S	E

مثال (32) : افترض ان برنامج باسكال مكتوب بالشكل التالي .. نفذ هذا الشكل واطهر النتائج (الإخراج) .

```
PROGRAM GHAZI (Output) ;
VAR A , B : INTEGER ;
    C , D : REAL ;
BEGIN
    A:=4 ;
    B:=6 ;
    C:=10.2 ;
    D:=25.4 ;
    Writeln (A,B:3,A+B:4) ;
    Writeln (A+9) ;
    Writeln (D);
END.
```

النتائج سوف تكون كما يلي :

4			6		0	1	0	0	E	+	0	1	
1	3		1			2	0						
2	5		4										

(5-9) كتابة وتصميم برنامج Pascal

من الممكن كتابة برنامج بسيط بلغة باسكال بعد أن وضعنا جمل الإدخال (القراءة) وجمل الإخراج (الطباعة) وصولاً إلى ادخال جمل جديدة في برنامج Pascal ، وقبل كتابة برامج بلغة Pascal لابد من معرفة الهدف النهائي المطلوب تحقيقه ، حيث نبدأ بعملية شرح وفهم البرنامج المطلوب كتابته من خلال كتابة مجموعة من الخطوات المنطقية الرئيسية المترابطة فيما بينها وهذا ما يعرف بالخوارزمية كما أشرنا في الفصل الثامن ونستخدم أحيانا أخرى ما يسمى بالمخطط الإنسيابي (Flowchart) والذي يوضح انسيابية البرنامج المهيكل ، ثم بعد كتابة وفهم هذا المخطط أو الخوارزمية نقوم بنقل هذا المخطط من صورته الحالية إلى

صورة برمجية طبقا للايعازات والخطوات المعروفة في جمل لغة Pascal ولنأخذ المثال التالي لتنفيذ ما سبق .

فلو فرضنا إننا نروم كتابة برنامج بلغة Pascal لإيجاد مساحة دائرة نصف قطرها معلوم ، فإن الخطوات المتسلسلة المنطقية والتي تسمى الخوارزمية التي نستطيع من خلالها الوصول إلى إيجاد مساحة الدائرة هي كما يلي :

الخطوة الأولى : ما هي المدخلات .

الخطوة الثانية : ما هي المخرجات .

فلو تفحصنا جيدا مثالنا اعلاه لوجدنا ان الخطوة الأولى سوف تشتمل على ما يلي:

1. نصف القطر والذي يمثل : Radius

2. النسبة الثابتة والتي تمثلها : Pi والتي تساوي 3.14154

3. استخدام قانون المساحة والتي يمثلها القانون $Pi * SQR(Radius)$

اما الخطوة اعلاه فتمثل الثانية تمثل في مثالنا هذا المساحة والتي هي : Area .

من خلال هذا الوصف فإن الخطوات الاساسية هي كما يلي :

أ) ادخال نصف قطر الدائرة .

ب) تعريف ثابت النسبة الثابتة PI

جـ) حساب المساحة .

د) طباعة النتيجة وهي المساحة .

ولهذا فإن هذا البرنامج سوف يكتب كما يلي :

```
1 [ PROGRAM Circle (Input,Output) ;
```

```
2 [ CONST Pi=3.14159 ;
```

```
3 [ VAR  
    R , AREA : REAL
```

```

4 { BEGIN
    READ (R);
    Area := Pi * SQR (R);
    Writeln (R,Area)
  } END.

```

حيث :

- 1- يمثل عنوان البرنامج .
 - 2 - يمثل تعريف الثوابت التي يجب تعريفها قبل تعريف المتغيرات .
 - 3- يمثل تعريف انواع المتغيرات .
 - 4- يمثل جمل لغة Pascal .
- إذن نستطيع القول أن أي برنامج بلغة Pascal يتكون من الاجزاء الرئيسية التالية:
- أولا : عنوان البرنامج .
- ثانيا : الإعلانات أو التصريحات أو التعريفات والتي تتكون من :
- أ- تعريف الثوابت .
 - ب- تعريف انواع المتغيرات .
 - ج - تعريف الاقترانات والاجراءات (سنأتي عليها لاحقا)
- ثالثا : جمل لغة Pascal
- نلاحظ من برنامجنا السابق ما يلي :

- 1- عنوان البرنامج يحتل سطرا واحدا ويتألف من كلمة PROGRAM ثم اسم البرنامج ثم اسم (ملف الادخال واسم ملف الاخراج) .
- 2- عرفنا أولا الثوابت (CONST) والمتمثلة بها النسبة الثابتة ثم بعد ذلك عرفنا المتغيرات والمتمثلة بنصف القطر ، والمساحة .
- 3- كتبنا برنامج Pascal محصورا بين جملي BEGIN , END ووضعنا نقطة بعد جملة END للدلالة على انتهاء البرنامج .

4- استخدمنا الفواصل (الفارزة المنقوطة) (;) بين أي جزئين ولم نستخدمها في الجملة الاحيرة .

5- من الممكن استخدام جمل توضيحية توضع بين نجمتين وقوسين وهذه لا تعتبر جمل Pascal وإنما تعتبر جمل توضيحية ويمكن ان نكتب برنامج Pascal السابق مع جمل توضيحية كما يلي :

PROGRAM Circle (Input,Output);

(* this PROGRAM is to Compute AREA of a circle *)

جملة توضيحية

CONST Pi=3.14159 ;

VAR Radius,Area : REAL ;

(*start the action statement *)

جملة توضيحية

BEGIN

(* read the radius *)

جملة توضيحية

READ (Radius);

Area := Pi * SQR (Radius);

(*print out *)

جملة توضيحية

WRITELN (Radius,Area)

END.

ولتنفيذ البرنامج السابق باستخدام حاسوب شخصي P. C فإننا نقوم بمتابعة الخطوات التالية :

1. بعد إدخال وطباعة البرنامج وظهوره على الشاشة نستخدم الايعاز Compile الذي يقوم بتدقيق البرنامج حسب قواعد لغة باسكال ثم تكوين ملف الهدف بلغة الالة .

2. بعد ذلك في حالة عدم وجود اية اخطاء نستخدم ايعاز التنفيذ : RUN .

3. ستظهر شاشة الإخراج (الشاشة السوداء) وحركة المؤشر في الزاوية اليسرى

إشارة الى وجوب إدخال مدخلات البرنامج أي قيمة نصف القطر Radius

فنقوم بإدخالها من لوحة المفاتيح ثم نضغط مفتاح الإدخال ENTER .

4. يتم تنفيذ البرنامج وتظهر النتيجة النهائية للمساحة .

مع ملاحظة ضرورة التدقيق على البرنامج لعدم حدوث أخطاء حيث يمكن ان

تكون هناك أخطاء ناتجة عن ترجمة البرنامج أو أخطاء تنفيذية ، أو اخطاء منطقية.

ومن الأخطاء الشائعة عن ترجمة البرنامج فهي : اخطاء الفواصل أو النقاط أو

الاقواس أو الإملاء أو استخدام رموز غير مقبولة أو استخدام ارقام تشابه الحروف

مثل حرف أو (O) ورقم (0) أو 1 و I ...

اما الاخطاء التنفيذية فهي : اخطاء كتابة في جمل القراءة او الطباعة أو اخطاء في

تعريف المتغيرات أو العمليات الحسابية .

اما الاخطاء المنطقية فهي : اخطاء ناتجة عن التكرار أو استخدام اقترانات بدون

كتابة اسماء ملفاتها في بداية البرنامج .

(9 - 6) جمل التحكم (السيطرة) Control Statement

ان جمل التحكم هي الجمل التي يتم استخدامها في برامج لغة Pascal

لغرض الانتقال من جملة إلى اخرى اي دون تنفيذ البرنامج بصورة متسلسلة

(Sequential) ، وذلك عند حاجة البرنامج لتنفيذ تقدم جملة على اخرى ، أو

من الممكن العودة إلى تنفيذ جملة سابقة منفذة (Executed) لغرض استخراج قيم

ترتبط بالجملة المنقول لها التنفيذ أو المطلوب العودة لها بالتنفيذ ، وهذا كله قبل

الانتقال إلى جمل جديدة تقع ضمن التابع المنطقي للبرنامج ، وفي بعض الأحيان

يحتاج البرنامج لتكرار تنفيذ مجموعة أو واحدة من الجمل والتعليمات لعدة مرات.

لهذا نستخدم جمل التحكم لمعالجة هذه الامور لتنفيذ عمليات متكررة قد تكون

مشروطة أو غير مشروطة . وقبل أن نستعرض الأساليب المختلفة المستخدمة في لغة

باسكال (PASCAL) للتحكم في سير خطوات البرنامج ، لابد من التطرق الى مفهوم الجمل المركبة .

(9-6-1) الجمل المركبة The Compound Statement

الجملة المركبة في لغة باسكال تتألف من مجموعة من جمل باسكال البسيطة التي يجب تنفيذها بشكل متتابع والصيغة العامة لها هي كما يلي :

```
BEGIN
    Statement (1);
    Statement (2);
    .....
    .....
    .....
    Statement (n-1);
    Statement (n)
END;
```

ونقصد بجمل باسكال البسيطة هي أي جملة تنتهي بفاصلة منقوطة ، ونلاحظ من الصيغة السابقة لجملة باسكال المركبة انها تبدأ بالكلمة المحجوزة BEGIN وتنتهي بالكلمة END تليها مباشرة فاصلة منقوطة (:) ، ويفصل بين كل جملة بسيطة وأخرى أيضاً فاصلة منقوطة ما عدا الجملة الأخيرة التي تأتي قبل كلمة END .

(9-6-2) جملة التحكم الشرطية IF – THEN – Statement

تسمى هذه الجملة في بعض الأحيان بنية اذا ...فإن ، وهي الجملة البنائية IF-THEN-Structure التي تستخدم عند تحقيق شرط محدد أو في بعض الأحيان عند عدم تحقيق شرط معين حيث يتم اختبار جملة الشرط والتي يعبر عنها (بتعبير منطقي) فإذا تحققت هذه الجملة يتم تنفيذ الجملة التالية لجملة THEN املا اذا لم تتحقق هذه الجملة فحينها يتم تنفيذ الجملة التي تأتي بعد جملة IF بالتسلسل والصيغة العامة لهذه الجملة ومنطق هذه الجملة موضحة كما يلي :

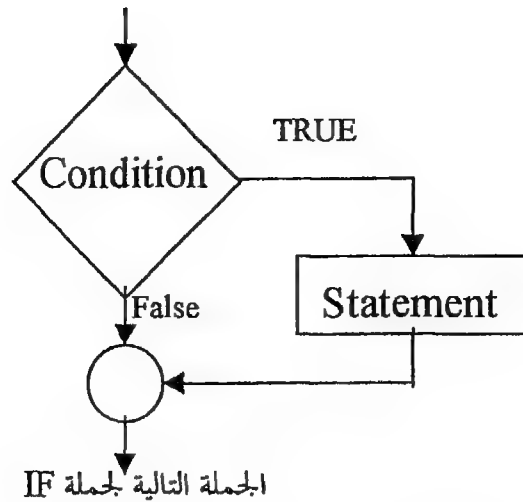
IF	Condition	THEN	Statement
----	-----------	------	-----------

حيث Condition هو تعبير منطقي.

Statement جملة باسكال بسيطة او مركبة والصيغة السابقة تعني انه اذا تحقق الشرط Condition فيتم الانتقال لتنفيذ جملة Statement التي تأتي بعد جملة THEN ، واذا لم يتحقق الشرط فيتم الانتقال إلى الجملة التالية لجملة IF الشرطية أي الجمل التي تلي الفاصلة المنقوطة في نهاية جملة IF ، والشكل (1) يوضح منطق عمل هذه الصيغة .

مثال (33): الايعازات التالية تمثل جملة الشرط بأنواعها وكما هو موضح لكل منها :

نوعها	الجملة
جملة شرطية حسابية	1. IF A<10 THEN B:=A+2 ;
جملة شرطية رمزية	2. IF Color <>'Y' THEN Color='G'
	3. IF(A<=0)OR(B<=12)OR(C<30)THEN BEGIN FLAG:=TRUE; Writeln('INVALID DATA') END;
جملة شرطية حسابية تحوي على اكثر من بديل لشرط واحد	
جملة شرطية تحوي على شرط رمزي وشرط حسابي	4. IF (A=200)OR(Sex='F') THEN Team := TRUE ;
جملة شرطية تحوي على شرط رمزي وشرط حسابي	5. IF (B>=0)AND(A='YES') THEN C>=(2*B-1) ;



الشكل (1) يمثل منطق عمل جملة IF-THEN

(3-6-9) جملة التحكم الشرطية

IF...THEN...ELSE... Statement

تسمى هذه الجملة في معناه إذا ...فإن... وإلا

وهي الجملة البنائية من جمل الشرط والتي تستخدم عند تحقق الشرط فيتم

تنفيذ الجملة الأولى وإذا لم يتحقق الشرط فيتم الانتقال إلى الجملة التي تأتي بعد

جملة الـ ELSE.. وصيغتها العامة على الصورة التالية:

IF Condition THEN Statement1 ELSE Statement2;

أو يمكن ان تكتب بالشكل التالي :

IF condition THEN
Statement1
ELSE
Statement2;

حيث condition تعبير منطقي Boolean Expression.

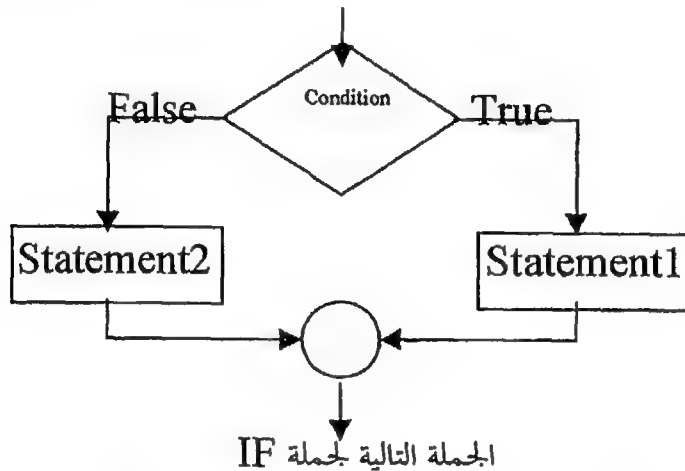
Statement1 جملة من جمل باسكال البسيطة أو المركبة .

Statement2 جملة من جمل باسكال البسيطة أو المركبة .

والصيغة السابقة تعني انه إذا كانت قيمة التعبير المنطقي محققة True فيتم مباشرة تنفيذ الجملة التي تأتي بعد جملة Then والتي هي Statement1 ، وإذا كان قيمة التعبير (False) فيتم تنفيذ الجملة التي تأتي بعد جملة ELSE والتي هي Statement2 والشكل (2) يوضح منطق هذه الجملة .

مثال(34):

```
IF A >= 0 THEN B:=A ELSE B:=-A;
Writeln (A:8:2 , B:8:2);
```



الشكل (2) منطق عمل جملة IF-THEN-ELSE

مثال(35):

```
IF FORM=SQR THEN
BEGIN
  READ (Side);
  Area := SQR(Side);
  Writeln ('Area of Sqr = ' , Area)
END
ELSE
BEGIN
  READLN (Base , Height);
  Area := Base * Hieght/2;
  Writeln ('Area of Triangle = ' Area);
End;
End;
```

في المثال أعلاه يتم تنفيذ المقطع الأول الذي يلي جملة THEN عندما يكون جواب الشرط صحيح وفي حالة عدم صحته يتم تنفيذ الجملة أو المقطع الذي يلي جملة ELSE

مثال (36) : الجمل التالية مقبولة بلغة باسكال :

```

1- IF A>=0 THEN Y:=2*A ELSE Y:=12 ;
2- IF FLIAT='I'THEN B='2'ELSE FIAT='C';
3- IF A>B THEN MAX:=A ELSE MAX:=B ;
4- IF X MOD 2 =0 THEN B=2 ELSE MAX:=B ;
5- IF T=1 THEN
    BEGIN
        READ (H,M);
        Area:=(H*M)/2 ;
    END ;
ELSE
    BEGIN
        READ (C,D,E);
        S:= (C+D+E)/2;
        Area:=SQRT(S*(S-A)*(S-B)*(S-C)) ;
    END;
WRITELN(Area);

```

(4-6-9) جملة التحكم الشرطية المتداخلة

Nested IF Statement

ونقصد بجملة التحكم الشرطية المتداخلة ، تداخل جملة IF مع بعضها لاكثر من مرة واحدة وهذه الجملة لها عدة اشكال متعددة منها ما يلي :

أ) الشكل الأول : والذي يسمى في بعض الأحيان بنية الطبقتين

Two-Layer Nesting

والصيغة العامة لهذا الشكل هو :

```
IF A1 THEN IF A2 THEN Statement1 ELSE Statement2
ELSE IF Conition3 THEN A3 ELSE Statement4 ;
```

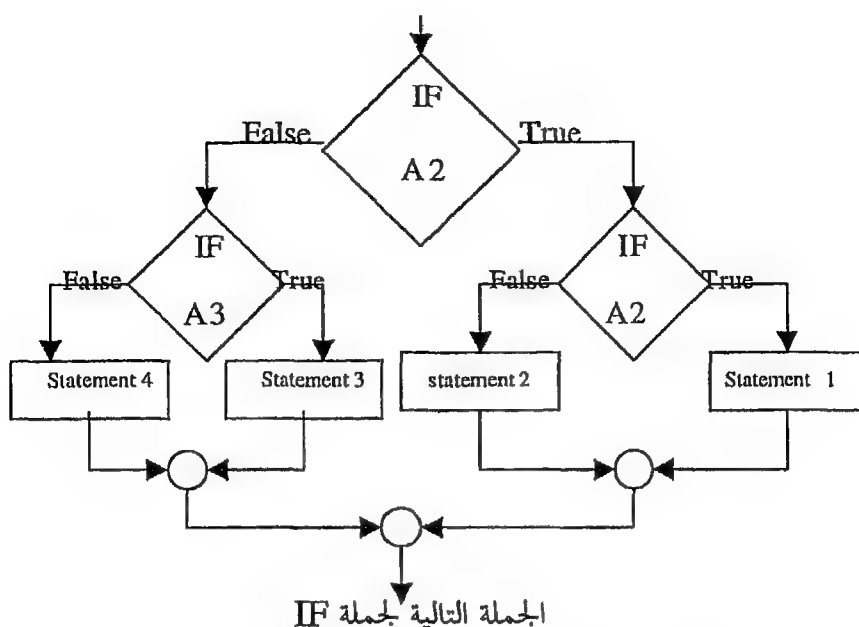
أو بشكل مفصل وأوضح كما يلي :

```
IF A1 THEN
    IF A2 THEN
        Statement1
    ELSE
        Statement2
ELSE
    IF A3 THEN
        Statement3
    ELSE
        Statement4 ;
```

حيث تمثل A1,A2,A3 جملة منطقية ، وتمثل Statement1, Statement2, Statement3
Statement4 , Statement3 جملة باسكال بسيطة أو مركبة :

وعمل هذه الجملة كما يلي :

يتم اختبار الشرط A1 فإذا كان الشرط متحقق (True) عندئذ يتم تنفيذ جملة
IF التي تأتي بعد THEN مباشرة (أي اختبار الشرط A2 فإذا كان متحقق يتم
تنفيذ Statement2 ، أما إذا كان الشرط A1 غير متحقق عندئذ يتم الانتقال الى
جملة IF التي تأتي بعد ELSE (أي اختبار الشرط A3 فإذا كان متحقق يتم تنفيذ
Statement3 ، وإذا كان غير محقق فيتم تنفيذ Statement4) . ونلاحظ من
خلال تركيب هذه الجملة انه يتم تنفيذ فقط إحدى جملة باسكال البسيطة أو
المركبة ، والشكل (3) يوضح مخطط سير العمليات لهذا النوع من جمل التحكم .



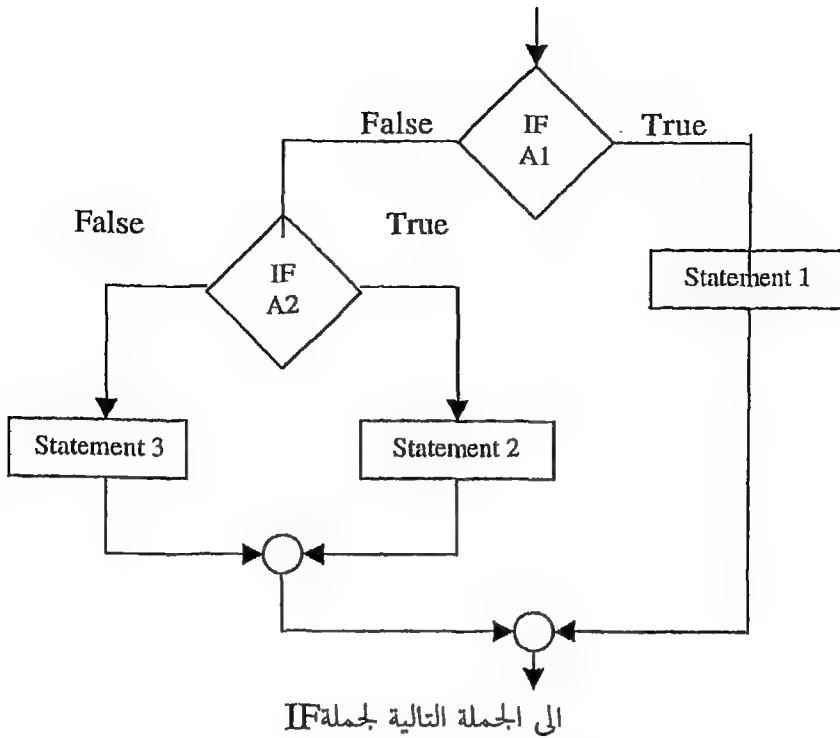
الشكل (3) منطق عمل جملة IF المتداخلة

ب) الشكل الثاني : والذي يحوي اكثر من بديل للشرط الثاني والصيغة العامة له:

```

IF A1 THEN
    Statement1
ELSE
    IF A2 THEN
        Statement2
    ELSE
        Statement3;
  
```

والشكل (4) التالي يوضح مخطط منطقي عمل هذه الصيغة .



الشكل (4) منطق عمل جملة IF المتداخلة

حيث A1.A2 هي تعابير منطقية , Statement1, Statement2 , Statement3 هي جملة باسكال بسيطة أو مركبة .

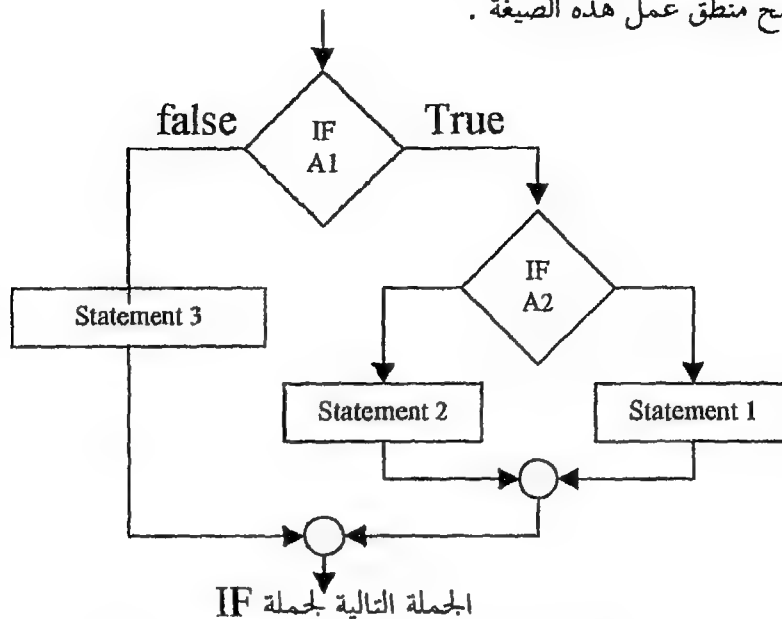
من الشكل (4) و الصيغة السابقة نلاحظ ان التداخل فقط في الجزء الثاني من جملة IF أي بعد جملة ELSE .

ج) الشكل الثالث : وهو الذي يتم التداخل فيه في الجزء الأول من جملة IF أي بعد جملة THEN والصيغة العامة لهذا الشكل هو :

```

IF A1 THEN
    IF A2 THEN
        Statement1
    ELSE
        Statement2
ELSE
    Statement3;
  
```


حيث A1 , A2 هي تعابير منطقية ، Statement1 ، Statement2 ، Statement3 ، والتي تحتل جمل باسكال البسيطة او المركبة . والشكل (5) يوضح منطق عمل هذه الصيغة .



الشكل (5) يوضح منطق عمل جملة IF المتداخلة

مثال(37) : البرنامج التالي هو لحساب قيمة الاقتران X المعرفة رياضياً حسب القيمة المؤشرة بجانب كل منها :

$$X = \begin{cases} Y^2 + 1 & ; Y > 0 \\ 0 & ; Y = 0 \\ -Y + 5 & ; Y < 0 \end{cases}$$

```

PROGRAM GHAZI (Input,Output);
VAR  X,Y : REAL
BEGIN
  READ (Y);
  IF Y > 0 THEN
    X:=SQR (Y) +1
  ELSE
    IF Y = 0 THEN
      X:=0.0
    ELSE
      X:=-Y + 5 ;
  WRITELN ('Y=' Y, ' ', 'X=',X:4:2 );
END.
  
```

مثال (38) : اكتب برنامج بلغة باسكال يقرأ عدد صحيح ثم يقوم بعملية اختبار هذا الرقم ويبين ان كان الرقم فردي أو زوجي .

```
PROGRAM GHAZI (Input,Output);
VAR
    Number , Rem : INTEGER ;
BEGIN
    READ (Number);
    REM:= Number MOD 2;
    IF Rem =0 THEN
        WRITELN('The Number',Number, 'Is EVEN')
    ELSE WRITELN('The Number',Number, 'Is Add')
End.
```

مثال (39) : اكتب برنامج بلغة باسكال يجد جذور المعادلة التربيعية من الدرجة

$$AX^2 + BX + C = 0 \quad ; \quad A \neq 0 \quad : \quad \text{الثانية التالية}$$

كما ان حل هذه المسألة يعتمد على قيمة المميز حيث ان $D = B^2 - 4AC$

فإذا كانت قيمة المميز موجبة فهناك جذرين حقيقيين ومختلفين وهما :

$$ROOT1 = \frac{-B + \sqrt{D}}{2A}$$

$$ROOT2 = \frac{-B - \sqrt{D}}{2A}$$

وإذا كانت قيمة المميز سالبة فإن للمعادلة جذرين غير حقيقيين

- إذا كانت قيمة المميز تساوي صفر فان للمعادلة جذر مضاعف

$$ROOT1 = ROOT2 = \frac{-B}{2A} \quad \text{يساوي:}$$

من خلال ما ذكر يمكن تنفيذ البرنامج لإيجاد جذور المعادلة التربيعية وكما يلي :

```

PROGRAM ABC (Input,Output);
(*Program calculates the real roots of a quadratic equation*)
VAR
    A,B,C,D          : REAL;
    Root1,Root2 : REAL;
BEGIN
    READ (A,B,C);
    Writeln('INPUT Data' :);
    Writeln('A=',A:8:2,'B=',B:8:2,'C=',C:8:2);
    IF A = 0.0 THEN
        BEGIN
            ROOT := - C / B ;
            Writeln('Root =' ,Root : 6 : 2 );
        END;
    ELSE
        D := B * B - 4.0 * A * C ;
        IF D >= 0.0 THEN
            BEGIN
                Root1:=(-B+SQRT(D)) / (2.0*A);
                Root2:=(-B-SQRT(D)) / (2.0*A);
                Writeln (' Root1= ',Root1:6:2);
                Writeln (' Root2 = ', Root2:6:2);
            END;
        ELSE
            Writeln ('No REAL Roots ');
        END.
END.

```

مثال (40) : مخزن تجاري يبيع بضاعة ويعطي خصومات على البضاعة وفق القاعدة التالية :

$$\text{Discount} = \begin{cases} 3\% \text{ OF } S & ; S < 500 \\ 6\% \text{ OF } S & ; 500 \leq S < 5000 \\ 9\% \text{ OF } S & ; S \geq 5000 \end{cases}$$

اكتب برنامج بلغة باسكال لحساب قيم الخصومات المعطاة على مجمل المبيعات

```

PROGRAM GHAZI (INPUT,OUTPUT);
VAR
    S,D : REAL;
BEGIN
    READLN (S);
    IF S<500 THEN
        D:=0.02*S
    ELSE
        IF S<5000 THEN
            D:=0.06*S
        ELSE
            D:=0.09*S;
    WRITELN ('FOR S=' , S:8:4);
    WRITELN ('DISCOUNT= ' ,D:8:4);
END.

```

CASE... Statement جملة التحكم الشرطية (5-6-9)

الصيغة العامة لهذه الجملة هي كما يلي :

CASE	Expression OF
	Value 1: statement 1;
	Value 2: statement 2;
	Value 3: statement 3;

	Value n: statement n ;
END.	

حيث :

Expression - تمثيل تعبير حسابي من النوع الصحيح أو يمكن ان يكون

تعبير رمزي أو منطقي .

value - هي قائمة الثوابت و تشمل كل منها قيمة أو مجموعة من القيم تفصل بينها الفارزة .

ومنطق عمل هذه الجملة كما يلي :

يتم اختبار قيمة التعبير Expression ، فإذا كانت value1 تحتوي قيمة هذا التعبير يتم تنفيذ الجملة Statement 1 ويتم الانتقال لتنفيذ الجملة التالية لجملة CASE-END ، وإذا كانت value2 تحوي قيمة هذا التعبير يتم تنفيذ الجملة Statement 2 ويتم الانتقال لتنفيذ الجملة التالية لجملة CASE-END. بمعنى آخر يتم تنفيذ جملة واحدة من مجموعة الجمل المشار إليها في الصيغة العامة حسب وجود قيمة التعبير في إحدى value .

أما إذا كانت قيمة التعبير Expression غير موجودة في أي من القيم المحصورة بين value 1 value n فعند ذلك لا ينفذ الحاسوب أي من الجمل الواردة في بقية جملة حالة الاختيار CASE .

مثال(41) : عند استخدام الإشارة الضوئية للمرور فإن اللون الأحمر هو (للتوقف) واللون الأصفر هو (الانتظار والتهوي) واللون الأخضر هو (للتجاوز أو المرور) والمقطع التالي من برنامج يستخدم جملة التحكم الشرطية CASE لحل المشكلة السابقة :

CASE color OF

‘Red’:WRITELN(‘STOP’);

‘Yellow’:WRITELN(‘WAIT’);

‘Green’:WRITELN(‘PASS’);

END;

حيث يتم اختيار إحدى الجمل أعلاه حسب قيمة اللون COLOR .

مثال (42) : أعطيت علامات الطلبة في جامعة العلوم بالشكل التالي :

التقدير	ادنى علامة
A	90
B	80
C	70
D	60
F	59

أكتب برنامج بلغة PASCAL لقراءة رقم الطالب ومعدله التراكمي وطباعة رقم الطالب وتقديره وفق الجدول اعلاه .

```
PROGRAM prog1 (INPUT,OUTPUT);
VAR
    StudNumber,Grade : INTEGER ;
    LetterGrade       : Char ;

BEGIN
    READLN (StudNumber , Grade );
    CASE Grade DIV 10 OF
        10 , 9 :LetterGrade := 'A';
        8      :LetterGrade := 'B';
        7      :LetterGrade := 'C';
        6      :LetterGrade := 'D';
        5 , 4,
        3 , 2,
        1 , 0 :LetterGrade := 'F';
    END;
    WRITELN(Student Number:9,Grade:3,
    LetterGrade);
END.
```

مثال (43) : المقطع التالي من برنامج لاستخدام جملة CASE لقيمة عددية

```
CASE (4*x + 5) OF
```

```
1:Y:=X-Z;
```

```
2:Y:=12;
```

```
4:Y:=-10;
```

```
5:Y:=sqr(x);
```

```
6:Y:=19;
```

```
END.
```

مثال (44) : كتابة برنامج يحسب مقدار علاوة الموظف حسب درجته بحيث اذا

كانت درجته تساوي :

- 0 ، أو 1 يمنح هذا الموظف علاوة تساوي صفرا .

- 2 , 3 , 4 يمنح هذا الموظف علاوة تساوي 30% من الراتب الاساسي .

- 5 , 6 , 7 يمنح هذا الموظف علاوة تساوي 60% من الراتب الاساسي .

- 8 , 9 يمنح هذا الموظف علاوة تساوي 90% من الراتب الاساسي

أكتب برنامج PASCAL باستخدام جملة CASE لتنفيذ ذلك ؟

```
PROGRAM SALARY (INPUT,OUTPUT);
```

```
VAR
```

```
    GRADE,BASICSALARY,MONTHSALARY:REAL;
```

```
BEGIN
```

```
    READ (GRADE , BASICSALARY );
```

```
    IF GRADE > 9 THEN
```

```
        WRITELN ('INVALID GRADE')
```

```
    ELSE
```

```
        BEGIN
```

```
            CASE GRADE OF
```

```
                0,1:B:=0 ;
```

```
                2,3,4:B:= 0.3 * BASICSALARY;
```

```
                5,6,7:B:= 0.6 * BASICSALARY;
```

```
                8,9 :B:= 0.9 * BASICSALARY;s
```

```
            END;
```

```
            MONTHSALARY := BASIC SALARY + B;
```

```
            WRITELN('MONTHSALARY=',MONTHSALAR);
```

```
        END.
```

(9-6-6) جملة الانتقال GOTO

تستخدم جملة الانتقال GOTO statement لنقل تسلسل تنفيذ البرنامج من جملة إلى أخرى عن طريق الانتقال غير المتسلسل ، أي الانتقال من التسلسل التتابعي إلى التسلسل المنطقي ، وهذه الجملة لا تستخدم إلا في الحالات الضرورية لأن وجودها في البرنامج يجعله أكثر صعوبة في القراءة والمتابعة والتصحيح وصيغة هذه الجملة العامة كما يلي :

GOTO N ;

حيث : N يمثل رقم الجملة التي يجب الانتقال إليها ، ويجب أن يكون هذا

الرقم معروف في بداية البرنامج باستخدام الكلمة المحجوزة Label .

مثال (45) : البرنامج التالي يوضح استخدام جملة GOTO .

```
PROGRAM GHAZI ( INPUT , OUTPUT );
```

```
  LABEL 10 ;
```

```
  CONST X = 2.05 ;
```

```
  VAR  A,B,C, : REAL ;
```

```
BEGIN
```

```
  10 READLN ( A,B,C );
```

```
  IF A= 0.0 THEN  GOTO  10
```

```
  B:= (A + B * C) / A ;
```

```
  WRITELN ('B=' , B;10:3) ;
```

```
END.
```

(9-7) جمل التكرار المشروط والتكرار باستخدام العدادات :

في الفقرة السابقة قمنا بشرح جمل التحكم التي تقوم بنقل سير تنفيذ العمليات من مكان إلى آخر استناداً إلى حالة الشرط ، وفي تلك الجمل لاحظنا أن جمل البرنامج تلك كان الحاسوب ينفذها مرة واحدة .

في موضوعنا الجديد الحالي باستخدام جمل التكرار المشروط أو التكرار باستخدام العدادات فإن جمل يتم تنفيذها لأكثر من مرة واحدة ، ويمكن التمييز بين

الانواع التالية من جمل التكرار :

- 1- استخدام جملة التكرار المشروط WHILE-DO
- 2- استخدام جملة التكرار المشروط REPEAT-UNTIL
- 3- استخدام جملة التكرار المتعدد FOR-NEXT
- 4- استخدام جملة التكرار المتداخل (Nested Loop Structure)

وسوف نقوم بشرح كل من تلك الانواع بشكل مفصل .

(9-7-1) استخدام جملة التكرار المشروط WHILE - DO :

أ) جملة التكرار المشروط WHILE-DO البسيطة :

في هذه الجملة نلاحظ ان عدد التكرارات التي تحدث لا يكون معروفاً مسبقاً وذلك لارتباط هذه الجملة بتحقيق أو عدم تحقق الشرط فإذا كانت قيمة الشرط صح True فسوف تستمر عملية التكرار إلى ان يصبح قيمة الشرط False خطأ ، فعندها يتم الانتقال إلى الجملة التالية لجملة WHILE وبنية هذه الجملة كما يلي:

WHILE	Condition	DO	Statement ;
-------	-----------	----	-------------

وهذه الجملة تعني انه مادام الشرط محقق فسيتم تنفيذ الجملة Statement الموجودة بعد كلمة DO ، ويستمر هذا التكرار إلى ان ينعدم تحقق الشرط ، فيتوقف التكرار ، وهذا النوع من الجمل التكرارية يدعى بالجمل التكرارية البسيطة والمثال التالي يوضح تركيب هذه الجملة :

مثال (46) :

.....

.....

Nmber:=0;

WHILE Number <=1000 DO

 Number:=Number+1;

.....

.....

نلاحظ في هذا المقطع من برنامج ما أن المتغير Number اخذ القيمة صفر ، وبعد عبارة WHILE وضعنا الشرط الذي يعني ما دامت قيمة المتغير أقل أو تساوي (1000) فإن الحاسوب سينفذ الجملة التي تلي عبارة DO أي ستصبح قيمة Number تساوي (1) ، ويعود إلى اختبار قيمة Number في الشرط وبما أنها أقل من (1000) فإنه سيقوم بتكرار تنفيذ الجملة التي تلي DO ، وهكذا يستمر بالتكرار إلى أن تصبح قيمة Number تساوي إلى (1001) عندئذ يتوقف الحاسوب عن التكرار ويتابع تنفيذ بقية جمل البرنامج التي تلي الفاصلة المنقوطة .

ب) جملة التكرار المشروط WHILE-DO المركبة :

في بعض الأحيان يتم استخدام جملة التكرار المشروط WHILE-DO بشكل مركب ، أي ان جملة حلقة التكرار تتكون من أكثر من جملة واحدة والبنية العامة لهذه الجملة كما يلي :

```

WHILE Condition DO
BEGIN
    Statement1;
    Statement2;
    Statement3;
    .....
    .....
    Statement n-1
    Statement n
END;

```

وهذه الجملة تعني أنه يتم أولاً اختبار الشرط (المنطقي) فإذا تحقق هذا الشرط أي أصبح TRUE يتم تنفيذ جملة حلقة التكرار (المركبة) اما اذا لم يتحقق الشرط فيتم الانتقال في التنفيذ إلى الجملة التالية لجملة WHILE وهكذا. والمثال التالي يوضح ما سبق :

مثال (47) : اكتب مقطع برنامج باسكال لحساب قيمة Y المعرفة رياضياً على الشكل التالي $Y=4X^2+X+7$ ولجميع قيم X المحصورة بين (-2, 2) أي $-2 \leq X \leq 2$ وبزيادة مقدارها 0.5 .

```
.....
.....
X:=-2.0;
WHILE X<=2.0 DO
BEGIN
    Y:=4*SQR(X) + X + 7.0;
    Writeln ( X:5:2,' ',Y:5:2);
    X:= X + 0.5
END;
```

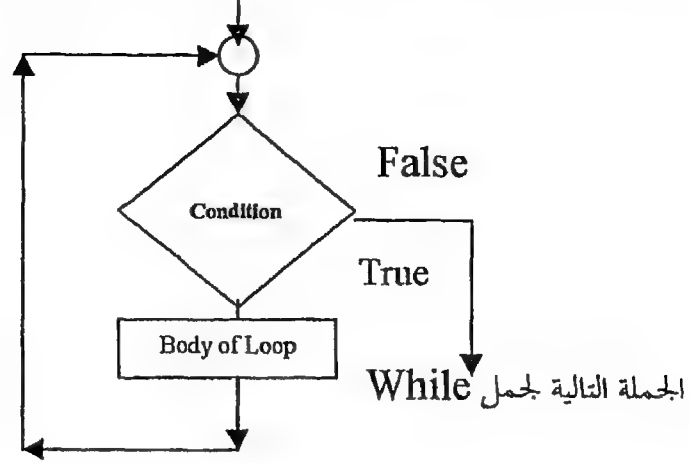
نلاحظ في هذا المقطع من البرنامج ان $X = -2$ في البداية ، وبعد عبارة WHILE وضعنا الشرط الذي يعني انه مادامت قيمة X اقل أو تساوي 2 فإن الحاسوب سينفذ مجموعة الجمل التي تلي عبارة DO والمحصورة بين عبارة BEGIN وعبارة END ، أي ان الحاسوب سيقوم بحساب قيمة Y حسب العلاقة المعطاه والتي تساوي (21) ثم يطبع على شاشة المخرجات ما يلي :

2.00	21.00
قيمة X	قيمة Y فراغ

ثم يحسب قيمة X الجديدة التي تساوي 2.5 ، وعندما يصل إلى عبارة END يعود إلى اختبار الشرط الذي يلي WHILE وبما انه متحقق يعود الحاسوب لينفذ الجمل بعد DO كما في المرة السابقة مع اخذ قيمة X الجديدة التي تساوي 2.5 وهكذا ... ويتوقف التكرار عندما تصبح قيمة X تساوي 2.5 وعندها يتابع الحاسوب تنفيذ الجمل التي تلي END;

بناءً على ما سبق نستطيع ان نعر عن جملة WHILE-DO بنوعيتها

البسيط والمركب ، بمخطط سير العمليات على الشكل التالي :



الشكل (6) منطق عمل جملة WHIL-DO

من هذا الشكل نلاحظ انه اذا كانت قيمة الشرط Condition هي False فيتم الانتقال إلى الجملة التالية لجملة WHILE ، واذا كانت قيمة الشرط Condition هي TRUE فيتم الانتقال لتنفيذ جملة التكرار Body of Loop بنوعيتها البسيط والمركب .

مثال (48) : اكتب برنامج بلغة باسكال لايجاد الوسط الحسابي لمجموعة من القيم المعلومة .

الحل : الوسط الحسابي نرمز له Average يحسب من العلاقة التالية :

$$\text{AVERAGE} = \frac{\text{SUM}}{N}$$

و SUM تمثل بمجموعة الأعداد والتي تبدأ X1 ، وتنتهي في Xn . أي ان :

$$\text{AVERAGE} = \frac{\text{SUM}}{N} = \frac{X1+X2+X3+X4+...+Xn}{N}$$

البرنامج :

```
PROGRAM Exam1 (Input,Output);
VAR
    N, Count          : INTEGER;
    SUM, X, AVERAGE: REAL;
BEGIN
    WRITELN ('Enter Number of Value : ');
    READLN(n);
    Sum:=0.0;
    Count:=1;
    WRITELN ('Enter Values : ');
    WHILE count <=n DO
    BEGIN
        READ (x);
        Sum:=sum+x;
        Count:=count+1;
    END;
    AVERAGE:=SUM / N;
    WRITELN ( ' THE AVERAGE=' ,AVEARAGE: 5:2);
END.
```

و عند تنفيذ البرنامج ستبدو شاشة المخرجات وكما يلي:

```
Enter number of value :5
Enter Value :
40
50
25
10
The AVERAGE = 30.00
```

مثال (49) : اكتب برنامج بلغة باسكال لإيجاد مضروب عدد حسب القاعدة التالية :

$$N! = \begin{cases} 1 & ; N = 0 \\ 1 * 2 * 3 * \dots * N & ; N \neq 0 \end{cases}$$

```
PROGRAM EXAM2 (Input,output);
VAR
    FACT,COUNT,N : INTEGER ;
BEGIN
    READLN (N) ;
    FACT:=1;
    COUNT:=1;
    IF N <>0 THEN
        WHILE COUNT<=N DO
            BEGIN
                FACT:=FACT*COUNT;
                COUNT:=COUNT +1;
            END;
        WRITELN ('FACTORIAL OF',N,'=',FACT :5);
        READLN;
    END.
```

فإذا أدخلنا قيمة N تساوي 5 عندئذ ستبدو شاشة المخرجات كما يلي :

5
FACTORIAL OF 5 = 120

مثال (50) : اكتب برنامج بلغة باسكال لإيجاد القيمة الصغرى من بين مجموعة قيم عددية صحيحة :

```
PROGRAM EXAM3 (Input,output);
VAR
    N,NUMBER,MIN,COUNT : INTEGER;
```

```

BEGIN
    MIN:=MAXINT;
    COUNT:=1;
    WRITELN ('Enter Number of Integer :');
    READLN(N);
    WRITELN ('Enter Values :');
    WHILE COUNT <=N DO
    BEGIN
        READLN (NUMBER);
        COUNT:=COUNT + 1;
        IF NUMBER < MIN THEN MIN:=X;
    END;
    WRITELN ('The Minimum Value =',MIN);
    READLN;
END.

```

عند تنفيذ البرنامج ستبدو شاشة المخرجات كما يلي :

```

Enter Number of INTEGER : 4
Enter Values :
78
45
30
85
the Minimum Value = 30

```

(9-7-2) جملة التكرار المشروط Repeat-Until

أ) جملة التكرار المشروط Repeat-Until البسيطة :

ان جملة التكرار المشروط Repeat-Until شبيهة بجملة التكرار المشروط While-Do حيث ان عدد التكرارات التي تحدث فيها لا يكون معروفاً وذلك لارتباط هذه الجملة بتحقق أو عدم تحقق الشرط الذي يتبعها ، فمنطق هذه الجملة بأنه يتم تنفيذ جملة التكرار ثم يتم اختبار الشرط فاذا لم يتحقق الشرط وقيمه False يستمر التكرار إلى أن يتحقق الشرط بحيث يصبح True عندها يتم

الانتقال إلى تنفيذ الجملة التالية لجملة Repeat والبنية العامة لهذه الجملة هي كما يلي :

REPEAT	Statement	UNTIL	Condition;
--------	-----------	-------	------------

يختلف تركيب هذه الجملة عن جملة WHELE-DO في انه يتم تنفيذ الجملة Statement على الاقل مرة واحدة ثم يتم اختبار تحقق الشرط بينما كما لاحظنا في الفقرة السابقة ان جملة WHELE-DO يتم اختبار تحقق الشرط أولاً ثم يتم تنفيذ الجملة Statement التي تأتي بعد DO . والمثال التالي يوضح تركيب هذه الجملة:

مثال (51) :

.....

.....

number :=0;

REPEAT

 number :=number+1 ;

UNTIL number > 1000;

.....

.....

نلاحظ في هذا المقطع من البرنامج أن المتغير number اخذ القيمة صفر في البداية، وبعد عبارة Repeat التي تعني كرر ، سينفذ الجملة التالية :

number := number +1

والتي ستصبح قيمة number تساوي (1) وبعد عبارة Until سيقوم بإختبار الذي يعني انه كرر الجملة السابقة حتى تصبح قيمة number اكبر من (1000)، وبما ان قيمة number تساوي (1) واقل من (1000) عندئذ سيقوم الحاسوب بتكرار جملة (number := number + 1) والتي ستصبح قيمة number تساوي (2)، وبعد عبارة Until سيتم اختبار الشرط وهكذا ...، إلى ان تصبح

قيمة المتغير number اكبر تماماً من 1000 وعندئذ سيتابع تنفيذ الجمل التي تلي الفاصلة المنقوطة .

(ب) استخدام جملة التكرار المشروط **Repeat-Until** (المركبة)
ومعنى هذه الجملة التكرارية المشروطة والمركبة هو انه يتم القيام بتكرار مجموعة من الجمل عدداً من المرات إلى ان يتحقق الشرط Condition ويكون جوابه True عندها يتوقف التكرار والبنية العامة لهذه الجملة هي كما يلي :

REPEAT Statement 1; Statement 2; Statement 3; Statement n; UNTIL Condition ;

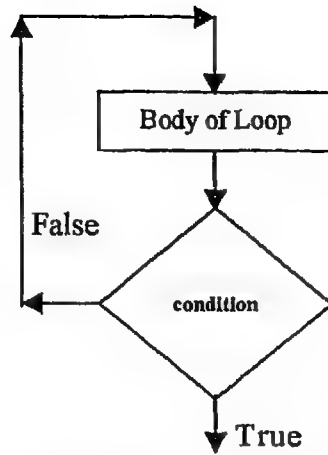
مثال (52): اكتب مقطع برنامج باسكال ينفذ الجملة **WRITELN (Count)** تسعة مرات ، حيث يأخذ العداد **Count** القيم 9,8,7,6,5,4,3,2,1 على التوالي.

```
.....  
.....  
count :=1;  
REPEAT  
    WRITELN (count);  
    count:=count +1;  
UNTIL count > 9;
```

نلاحظ في هذا المثال ان العداد **count** يأخذ القيمة (1) ويزداد قيمته أيضاً بمقدار واحد (1) في كل مرة حتى يتحقق الشرط فيصبح العداد **count** يساوي أو أكبر من 9 فيتوقف عمل التكرار .

وبناء على ما سبق نستطيع أن نعبر عن جملة **REPEATUNTEL**

البسيطة والمركبة بمخطط سير العمليات التالي:



الى الجملة التالية لجملة REPEAT

الشكل (7) يمثل منطق عمل جملة (REPEAT-UNTIL)

حيث نلاحظ من هذا الشكل (7) أنه يتم الاختبار للجملة بعد تنفيذ جملة حلقة التكرار وهذا يعني أن جملة Repeat ينفذ التكرار مرة واحدة على الأقل قبل الانتقال إلى الجمل التالية في البرنامج .

مثال (53) : أعد حل المثال (48) باستخدام جملة التكرار المشروط -REPEAT UNTIL

PROGRAM EXAM5 (INPUT ,OUTPUT)

VAR

N,count : INTEGER;

sum,AVERGE :REAL;

BEGEN

sum:=0.0;

count:=1;

WRITELN ('Enter Number of value :');

READLN (N);

WRITELN ('Enter Values:');

REPEAT

READLN (X);

```

        sum:=sum + x;
        count:= count + 1;
    Until count > N ;
    AVERAGE := sum / N ;
    Writeln ('THE AVERAGE= ',AVERAGE :5:2);
END.

```

عند تنفيذ البرنامج ستبدو شاشة المخرجات كما يلي :

```

Enter Number of Value :5
Enter Values :
40
50
25
25
10
THE AVERAGE = 30.00

```

مثال (54) : أعد حل المثال (49) باستخدام جملة التكرار المشروط - REPEAT- UNTIL .

```

PROGRAM EXAM6 (INPUT.OUTPUT);
VAR
    FACT,COUNT,N: INTEGER ;
BEGIN
    Writeln ('Enter Number:');
    Readln (N);
    FACT:=1;
    COUNT:=1;
    IF N <> 0 THEN
        REPEAT
            FACT:= FACT * COUNT ;
            COUNT:= COUNT + 1;
        UNTIL COUNT > N ;
    Writeln ('FACTORIAL of ',N,'=',FACT :5);
    Readln;
END.

```

وعند تنفيذ البرنامج ستبدو شاشة المخرجات كما يلي :

```
Enter number :5
FACTORIAL of 5 = 120
```

مثال (55) : اكتب برنامج بلغة باسكال لإيجاد أكبر قيمة من قائمة تحوي على N قيمة عددية صحيحة .

اذن المطلوب في هذا البرنامج هو إيجاد القيمة العظمى Maxint .
البرنامج :

```
PROGRAM EXAM7(Input,Output);
VAR
    N,X,largest,count : INTEGER ;
BEGIN
    largest:=MAXINT;
    count:=0;
    WRITELN ('Enter Number of Value :');
    READLN('N');
    WRITELN('Enter Value :');
    REPEAT
        READ (X);
        count:=count +1;
        IF X > largest THEN largest :=X;
    UNTIL count=N;
    WRITELN ('The Largest Value=',largest);
    READLN;
END.
```

وعند تنفيذ البرنامج ستبدو شاشة المخرجات كما يلي :

```
Enter Number of Value 5
Enter Values :
65
87
98
0
94
The Largest Value = 98
```

(3-7-9) جملة التكرار FOR

نستخدم هذه الجملة لتكرار تنفيذ مجموعة من جمل باسكال عدداً معروفاً ومحددًا من المرات ، والبنية العامة لهذه الجملة :

FOR Counter := Initial Value TO Final Value DO Statement;

أو

FOR Counter:=Initial Value DOWNTO Final Value DO Statement;

حيث

Counter	اسم العدد لمتغير من نوع صحيح
Initial Value	القيمة الابتدائية للعدد
Final Value	القيمة النهائية للعدد
To	تستخدم في حالة العد التصاعدي
	اي عندما Initial Value < Final Value
DOWNTO	تستخدم في حالة العد التنازلي
	اي عندما Initial Value > Final Value
Statement	الجملة التي تحتاج إلى التكرار

ملاحظة : الجملة Statement التي تحتاج إلى التكرار عدداً من المرات يمكن ان تكون بسيطة (أي جملة واحدة) ويمكن ان تكون مركبة (عدة جمل) وفي هذه الحالة يجب أن تبدأ الجملة المركبة بعبارة BEGIN وتنتهي الجملة بعبارة END;

أي تأخذ البنية العامة الشكل التالي :

```
FOR Counter := initial value TO final value DO
BEGIN
    Statement 1;
    Statement 2;
    :
    :
    Statement n;
END;
```

مثال (56) : اكتب مقطع برنامج باسكال يجمع قيم الأعداد من 1-9 .
في حالة استخدام جملة TO التصاعدية يكون المثال كما يلي :

```
.....  
.....  
SUM:=0;  
FOR COUNT:=1 TO 9 DO  
SUM:=SUM+COUNT;  
.....
```

وفي حالة استخدام جملة DOWNTO التنازلية فيصبح المثال السابق كما يلي :

```
.....  
.....  
SUM:=9;  
FOR COUNT:=9 DOWNTO 1 DO  
SUM:=SUM+COUNT;  
.....  
.....
```

وعمل هذا الجزء من البرنامج كما يلي :

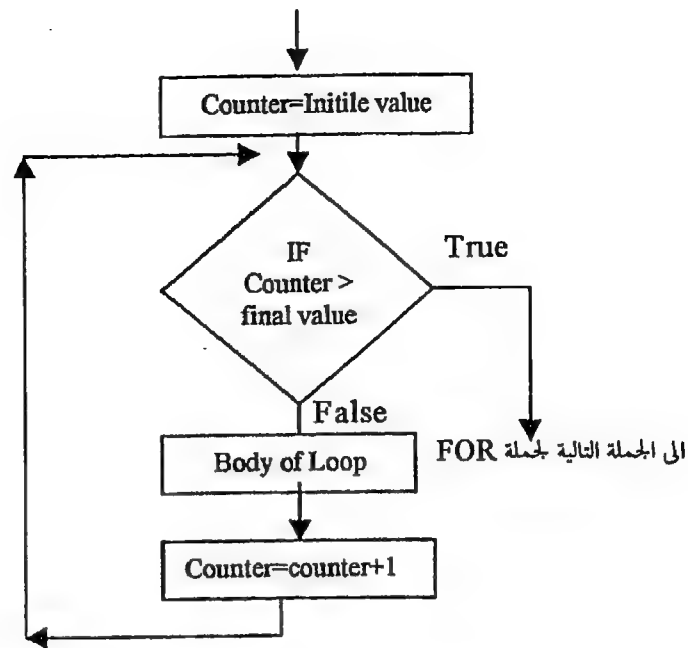
- يبدأ عمل العداد بمقدار القيمة الابتدائية المعروفة في جملة التكرار .
- يتم مقارنة القيمة الجديدة للعداد بالقيمة النهائية فإذا كانت القيمتين متساويتين يتم التوقف وإذا كانت قيمة العداد الأولية اقل من النهائية يتم تكرار تنفيذ الجملة التي تأتي بعد عبارة DO علماً أن قيمة العداد في حالة استخدام القيمة التصاعدية يتم إضافة واحد (1) وفي حالة استخدام القيمة التنازلية يتم إنقاص (1) وهكذا .

والشكل (8) يمثل منطق عمل الجملة FOR

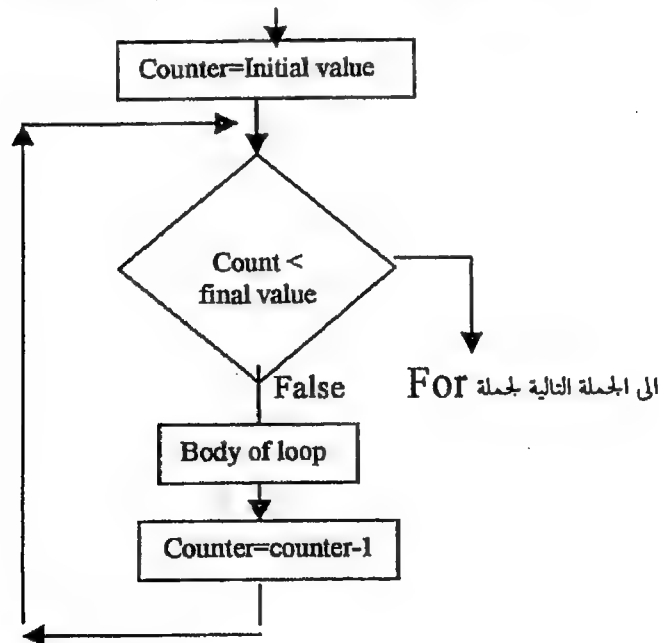
في حالة : $\text{Initial Value} < \text{Final Value}$ أي ان القيمة الابتدائية للعداد اصغر من القيمة النهائية .

والشكل (9) يمثل منطق عمل الجملة For

في حالة : $\text{Initial Value} > \text{Final Value}$ أي القيمة النهائية اكبر من القيمة النهائية .

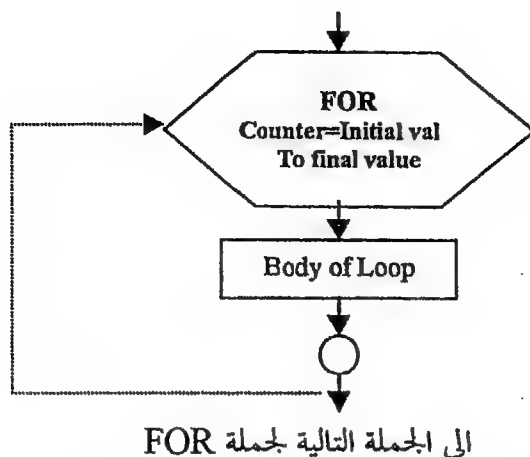


والشكل (8) اعلاه يمثل منطق عمل FOR-TO



الشكل (9) منطق عمل جملة FOR-DOWNTO

وعند رسم مخططات سير العمليات يعبر عن جملة FOR كما هو موضح في الشكل (10) :



الشكل (10) مخطط سير العمليات لجملة FOR

مثال (57) : اعد حل المثال (48) السابق باستخدام جملة التكرار FOR .

```

PROGRAM EXAM9(Input,Output);
VAR
    Count,N : INTEGER
    Sum,Average,X : REAL;
BEGIN
    WRITELN('Enter Number of Value:');
    READLN(N);
    Sum := 0.0;
    WRITELN('Enter Values :');
    FOR count := 1 TO n Do
    BEGIN
        READLN(X);
        Sum:=Sum+1;
    END;
    Average := Sum/N;
    WRITELN('The Average = ', Average :5:2);
END.
    
```

وعند تنفيذ البرنامج ستبدو شاشة المخرجات كما يلي :

Enter Number of Value : 5

Enter Values

40

50

25

25

10

The Average = 30.00

مثال (58) : اعد حل المثال (49) السابق باستخدام جملة التكرار FOR

```
PROGRAM EXAM10(Input,Output);
```

```
VAR
```

```
    FACT,COUNT,N : INTEGER;
```

```
BEGIN
```

```
    WRITELN('Enter The Number :');
```

```
    READLN (N);
```

```
    FACT:=1;
```

```
    FOR COUNT:=1 TO N DO
```

```
    BEGIN
```

```
        FACT := FACT * COUNT;
```

```
    END;
```

```
    WRITELN ('FACTORIAL Of 'N,' ' ,FACT :5);
```

```
    READLN;
```

```
END.
```

وعند تنفيذ البرنامج ستبدو شاشة المخرجات كما يلي :

Enter The Number : 6

FACTORIAL Of 6 = 720

Nested Repetition Structure (4-7-9) جمل التكرار المتداخلة

نستخدم هذه الجملة في الحالات التي يحوي فيها البرنامج على أكثر من

حلقة تكرارية متداخلة أي أننا نقوم بوضع بنية تكرارية داخل بنية تكرارية أخرى

وهذا يؤدي إلى تكرار عمليات مكررة سابقة شريطة ان تكون الجملة التكرارية الثانية واقعة داخل بنية جملة التكرار الخارجية وبتفصيل اكثر نوضح هذا التكرار من خلال المخطط في الشكل (11) :



والشكل (11) يمثل منطق عمل جملة FOR المتداخلة

ولنوضح ما سبق من خلال المثالين التاليين الأول لكتابة برنامج بدون تداخل والثاني باستخدام تداخل جمل التكرار .

ملاحظة : ان التداخل يمكن ان يتم باستخدام جملة التكرار WHILE-DO أو جملة التكرار REPEAT-UNTIL أو جملة التكرار FOR أو خليط بينهما وهذا يتعلق بالبرمج أولاً وبنوع خوارزمية الحل ثانياً .

مثال (59) : اكتب برنامج بلغة باسكال لحساب معدل اعمار طلبة احدى الشعب في جامعة معينة .

الحل : نفرض ان عدد الطلاب هو 60 طالب .

اذن المطلوب قراءة اعمار 60 طالب ثم يتم جمعها في Counter ويطلق عليه اسم المجموع sum ، ثم تتوقف العملية بعد قراءة الـ (60) عمر ونحزنها في sum ، ثم يتم ... من حلقة التكرار ، ثم يتم تقسيم مجموع

الاعمار الـ(60) على عدد الطلاب (60) وذلك لحساب الوسط الحسابي (المعدل) للعمر ثم يتم طبع قيمته وتتوقف .

نرمز لعمر الطالب بـ A ، وبمجموع اعمار الطلبة في الشعبة بـ sum ، ومعدل العمر بـ Mean

```
PROGRAM EXAM11(Input,Output);
VAR
    COUNT : INTEGER;
    A,SUM,MEAN:REAL;
BEGIN
    SUM:=0.0;
    FOR COUNT:=1 TO 60 DO
        BEGIN
            READ (A);
            SUM:=SUM+A;
        END;
    MEAN:=SUM/60;
    Writeln ('Mean =', MEAN);
END.
```

تمثل
حلقة
تكرار

مثال (60) : اكتب برنامج بلغة باسكال لحساب معدل اعمار الطلبة في كل شعبة من جامعة ما مع العلم ان الجامعة تضم عشرة شعب وكل شعبه تضم 60 طالباً .

الحل : ان هذا البرنامج هو تعميم للبرنامج المعروض في المثال (11) ، إذا اننا سنقوم بتكرار حلقة التكرار في المثال (11) عشرة مرات ، فإذا رمزنا بـ CLASS للعدد الذي سيستخدم لعد الصفوف العشرة ، فإن البرنامج يأخذ الشكل التالي :

```
PROGRAM EXAM12(Input,Output);
VAR
    CLASS,COUNT:INTEGER;
    A,SUM,MEAN : REAL;
```

```

BEGIN
FOR CALSS:=1 TO 10 DO
  BEGIN
    SUM:=0.0;
    FOR COUNT :=1 TO 60 DO
      BEGIN
        READ (A);
        SUM :=SUM+A;
      END;
      MEAN:=SUM/60;
      Writeln('Class=',CLASS,'Mean=',MEAN);
    END;
  END;
END.

```

البنية الخارجية 10 مرات

البنية الداخلية 60 مرة

مثال (61) : اعد كتابة البرنامج في المثال (60) باستخدام REPET-UNTIL

```

PROGRAM EXAM13(Input,Output);
VAR
  CLASS,COUNT :INTEGER;
  A,SUM,MEAN : REAL;
BEGIN
  CLASS=1;
  REPEAT
    COUNT := 1;
    SUM := 0.0;
    REPEAT
      READ(A);
      SUM:=SUM+A;
      COUNT := COUNT + 1;
    UNTIL COUNT > 60;
    MEAN:=SUM/60;
    Writeln('Class=',CLASS,'Mean=',MEAN);
    CLASS=CLASS+1;
  UNTIL CLASS > 10 ;
END.

```

مثال (62) : اعد كتابة البرنامج في المثال (12) باستخدام WHILE-DO (يترك للطالب) .

مثال (63) : اكتب برنامج بلغة باسكال لحساب قيمة العدد النيري e حسب العلاقة التالية :

$$e = 1 + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{N!}$$

مستخدما التكرار المشروط WHILE-DO لحلقة التكرار الخارجية وجملة FOR لحلقة التكرار الداخلية .

الحل : اسماء المتغيرات التي ستستخدم في البرنامج هي :

N عدد الحدود

COUNT1 عداد لحدود العلاقة

COUNT2 عداد لاستخدامه عند حساب مضروب الأعداد في المقامات

SUM يجمع لتجميع حدود العلاقة

FACT لوضع فيه مضروب عدد

PROGRAM EXAM15 (Input,Output);

VAR

N,COUNT1,COUNT2,FACT :INTEGER;

SUM :REAL;

BEGIN

WRAITELN(Enter THE Number of Terems:');

READLN (N);

COUNT1 :=1;

SUM:=0.0;

WHILE COUNT1 <=N DO

BEGIN

FACT := 1;

FOR COUNT2 :=1 TO COUNT1 DO

FACT:=FACT*COUNT2;

```

SUM := SUM + 1.0 / FACT;
COUNT1:= COUNT1 + 1;
END;
WRITELN ('e=',SUM:4:2);
END.

```

وعند تنفيذ البرنامج تبدو شاشة المخرجات كما يلي :

```

Enter The Number of Terms : 5
E = 1.72

```

مثال (64) : اكتب برنامج بلغة باسكال لطباعة الحرف X على خمسة اسطر بالشكل التالي :

في السطر الأول حرف واحد

في السطر الثاني حرفان

في السطر الثالث ثلاثة احرف

في السطر الرابع اربعة احرف

في السطر الخامس خمسة احرف

```

PROGRAM EXAM16 (Input,Output);
VAR
    ROW,LITER : INTEGER;
BEGIN
    FOR ROW := 1 TO 5 ROW
    BEGIN
        FOR LETTER:=1 TO ROW DO
            WRITELN('X');
        WRITELN;
    END;
END.

```

وعند تنفيذ البرنامج ستبدو على الشاشة كما يلي :

```

X
XX
XXX
XXXX
XXXXX

```

تمارين وأسئلة عامة

1- افترض ان

VAR

A,B,C : INTEGER ;

X,Y,Z : REAL ;

وضح أي من الجمل التالية مقبولة أو غير مقبولة في لغة باسكال كجمل اسناد .

1. $X := B * (C + Y) / Y$
2. $B := (A + C) \text{ DIV } 2$
3. $A := ((3 * B) \text{ MOD } (C \text{ DIV } 2)) * 20.0$
4. $A = \text{TRUNC} (((4 * C) \text{ MOD } (C \text{ DIV } 2) + 10.2))$
5. $Y := X * X + X + 3 * 3 * \text{SQR}(Y)$

2- اكتب الجمل اللازمة للاعلان عن نوع كل من المتغيرات التالية :

<u>المتغير</u>	<u>نوعه</u>
1. ROOT	حقيقي
2. X	حقيقي
3. Count	صحيح
4. Color	رمزي
5. State	منطقي
6. Index	صحيح

3- افترض ان قيمة a هي الصواب True

b هي الخطأ False

c هي الصواب True

جد قيمة كل من التعبيرات المنطقية التالية

1. $a \text{ AND } (b \text{ OR } c)$
2. $(a \text{ AND } b) \text{ OR } (a \text{ AND } c)$
3. $\text{NOT } a \text{ OR } \text{NOT } c$
4. $a \text{ AND } \text{NOT } b \text{ OR } \text{NOT } c$

4- جد قيمة التعابير الحسابية التالية :

1. Round (-2.8)
2. Round (-5.1)
3. Trunc (-2.8)
4. Trunc (-5.1)
5. Trunc (3.9)
6. Round (3.9)
7. ABC (3)
8. SQR (-2)

5- حدد نوع التعبير لكل مما يلي :

1. $6*3+14/2$
2. Counter
3. $SQR(Z+4)/6$
4. $Minimum=200$
5. $Root \leq 2.0$

6- اكتب التعابير التالية بلغة باسكال :

1. $(Z+A) (B+A)$
2. $6(M-Y)$
3. QRZ^2+5YX
4. $X+CY^{-2}$
5. $Z^3 / 3!$

7- اكتب التعابير التالية بلغة باسكال بالاستعانة بالاقترانات المكتبية القياسية :

1. $|C+D| GM$
2. $\sqrt{A^2 + C^2}$
3. $\cos(A+B)$
4. $TAN^{-1} (C/D)$
5. E^{c+b}

8- اكتب المعادلات التالية بلغة باسكال

1. $G=(C^2 / 2!) + (C^3 / 4!)$
2. $H= (X^8R-ST^4) / (aX+b)$
3. $D= (1+L)^{1/2}$
4. $L:= \sqrt{X-4X+5} / (4(X-1))$
5. $M:=(R e^{9Ln(X)} - St^4) / (a(x+b))$

9- نفرض ان :

VAR

A,B,C : REAL;

I,J,K : INTEGER;

وبفرض ان سجلات ملف البيانات المدخلة هي :

44	5.56	16	14
5	17		21
19	11		

والمطلوب ما هي نتائج تنفيذ جمل القراءة التالية :

READ(I,A,B);

READLN(C);

READLN(K);

READ (J);

10- بفرض ان :

X=5.6,Y=2 ,Z=3.0

فما هي نتائج تنفيذ الحاسوب للجمل التالية :

WRITELN(X,Y,Z);

WRITELN;

WRITELN(X,'**',Y,'**',Z);

WRITE(X:6:2);

WRITELN(Z:5:2);

WRITELN;

WRITELN('*****');;

12- اكتب برنامج بلغة باسكال لادخال ثلاثة أعداد وطباعتهم ، وطباعة العدد

الاكبر من بين الأعداد الثلاث .

13- اكتب برنامج بلغة باسكال لايجاد جذور معادلة من الدرجة الثانية :

$$AX^2 + BX + C = 0$$

14- اكتب برنامج باسكال لحساب قيمة عمولة البائع المحصلة على مجمل مبيعاته
S حسب القاعدة التالية :

$$\text{Commission} = \left\{ \begin{array}{ll} 0.02 \text{ OF } S & ; S < 500 \\ 0.05 \text{ OF } S & ; 500 \leq S \leq 5000 \\ 0.10 \text{ OF } S & ; S > 5000 \end{array} \right\}$$

15- اكتب برنامج بلغة باسكال لحساب الاقتران التالي :

$$F(X) = X^2 + 2X^3 + X - 7$$

لجميع قيم X من 1 إلى 10 وزيادة دورية مقدارها 0.25 .

16- تقدم 40 طالب لفحص في مادة المقدمة و سجلت العلامات ، والمطلوب
كتابة برنامج باسكال لايجاد عدد الطلاب الناجحين وعدد الطلاب الراسيين
وعدد الطلاب اللذين تزيد علاماتهم عن 80 .

17- اكتب برنامج لحساب الاقتران التالي :

$$F(X) = \left\{ \begin{array}{ll} 1 & ; X < 0 \\ 4X^2 + 1 & ; 0 \leq X \leq 1 \\ 5X^3 + 3X + 2 & ; X > 1 \end{array} \right.$$

18- اكتب برنامج بلغة باسكال لطباعة وحساب مجموع الأعداد الزوجية من 2
إلى 50 .

19- اكتب برنامج بلغة باسكال لحساب كلا مما يلي :

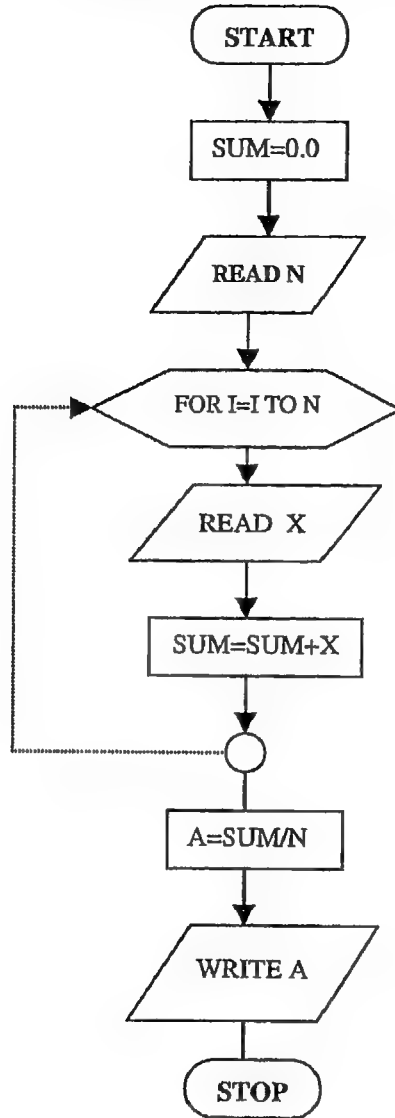
$$1) \sin(X) = \frac{X}{1!} - \frac{X^3}{3!} + \frac{X^5}{5!} - \dots$$

$$2) \pi = 4(1 - 1/3 + 1/5 - 1/7 + 1/9 - \dots)$$

$$3) e^x = 1 + X/1! + X^2/2! + X^3/3! + \dots + X^N/N!$$

20- اكتب برنامج بلغة باسكال لايجاد اكبر عنصر واصغر عنصر من عناصر
مصفوفة A .

21- اكتب البرنامج بلغة باسكال المكافئ للمخطط التالي :



22- اكتب برنامج لايجاد اكبر عنصر في كل سطر من عناصر مصفوفة ما

23- اكتب برنامج لايجاد اصغر عنصر في كل عامود من اعمدة مصفوفة ما

24- اكتب برنامج كل جملة معادلات باستخدام طريقة غوس-جوردان

25- اكتب برنامج ترتيب مجموعة من القيم المدخلة بشكل عشوائي ترتيباً تصاعدياً .

26- اكتب برنامج لحساب معدل علامات الطلاب المتقدمين لامتحان المقدمة

ومقدار انحراف كل علامة عن المعدل : مع العلم ان

بمجموع علامات الطلبة

$$\frac{\text{المعدل} = \text{معدل}}{\text{عدد الطلبة}}$$

عدد الطلبة

انحراف كل علامة عن المعدل = علامة الطالب - المعدل

على ان تبدو شاشة المخرجات بالشكل التالي :

AVERAGE =	
العلامة	الانحراف
<u>SCORE</u>	<u>DEVIATION</u>
.....
.....

الفصل العاشر

البرمجة بلغة باسكال

PASCAL PROGRAMMING

بيانات المستخدم التعريفية — المصفوفات — البرامج الفرعية

السجلات والملفات

(1-10) بناء أنواع بيانات جديدة في لغة باسكال

تطرقنا في ما مضى الى الأنواع الأساسية من البيانات القياسية مثل :

Boolean data , char data , Real data , Integer data

ولكن هناك إضافة إلى ذلك أنواع من البيانات التي يمكن تسميتها ببيانات المستخدم التعريفية **User – defined – data** : وهذه المعطيات (البيانات) تكتب بطرق خاصة بحيث يمكن تحديد القيم التي يمكن أن تأخذها هذه الأنواع ثم يتم تعريف اسمها جماعيا لهذا النوع ثم الإعلان عن متغيرات هذا النوع بحيث تأخذ هذه المتغيرات جزء أو كل من القيم التي سبق تحديدها وهذه الأنواع يمكن تقسيمها الى قسمين رئيسيين هما :

أ) البيانات معدودة القيم (التعددية)

ب) البيانات ذات المجال الجزئي (الجزئية)

(1-1-10) البيانات معدودة القيم Enumerated-Type data

هذا النوع من البيانات يتكون كما أوضحنا من مفردات من نوع واحد وتحمل اسما جماعيا يدل عليها وصورتها العامة هي :

TYPE name (data1, data2, ... , data n) ;

حيث Name يمثل اسم نوع البيانات الجديد .

تمثل قائمة القيم التي يأخذها هذا النوع الجديد من البيانات .

data 1
.....
data n

مثال :

TYPE Months=(January, February, March, April, May, June, July, August, September, October, November, December);

حيث نلاحظ في مثالنا هذا أن الشهر الممثل Months يمثل الاسم الجماعي
الواصف لنوع البيانات TYPE وتكون هذه المفردات للبيانات مرتبة بشكل
متسلسل وبالتالي يمكن مقارنتها ببعضها باستخدام إشارات الأصغر والأكبر
والعديد من إشارات المقارنة، كذلك يمكن أن نصرح عن المتغيرات من هذا النوع
كما يلي:

VAR

Y: Months ;

وبالتالي يمكن إسناد أي من القيم المعرفة في Months للمتغير Y أي يمكن ان
نكتب في البرنامج ما يلي ; Y:=March ، والمثال التالي يوضح استخدام
إشارات العمليات المنطقية :

القيم المنطقية	التعبير المنطقي
True	March < April
True	November > = July
True	June < > July
True	SUCC (January) = May
True	PRED (September) = August

حيث استطعنا أن نستخدم بعض الاقتارات القياسية في لغة Pascal مثل
SUCC و PRED مع البيانات القيم . وبما أن هذه البيانات مرتبة بشكل
متسلسل فإنها تأخذ أرقام تصاعديّة تبدأ بالصفر وتنتهي بـ n-1 لآخر مفردة وكما
في المثال التالي:

ord (January) = 0

ord (March) = 2

ord (July) = 6

ord (December) = 11

حيث تمثل المفردة الأولى الشهر الأول من السنة والثانية تمثل الشهر الثالث من السنة والمفردة الثالثة تمثل الشهر السابع من السنة والمفردة الأخيرة تمثل الشهر الثاني عشر من السنة .

مع العلم انه في بعض الأحيان يمكن أن نستخدم في قيم الأنواع التعددية جملة من جمل التكرار For مثال :

FOR Months := January TO December DO ...
--

أكتب برنامج بلغة باسكال لطباعة أيام الأسبوع باستخدام الأنواع (المعطيات) المعدودة القيم

```
PROGRAM DAYS (INPUT,OUTPUT);
TYPE Days_Of_Week=(Sat,Sun,Mon,Tus,Wen,Thur,Frid);
VAR
    Day:Days_Of_Week ;
BEGIN
    FOR Day := Sat OF Frid DO
        WRITELN (day);
    END.
```

(10-1-2) البيانات الجزئية Subrange-Type-data

أعطيت اسم الجزئية كونها تمثل جزء من سلسلة متكاملة من مفردات المعطيات البسيطة مثل البيانات (الرمزية او المنطقية او الصحيحة) او البيانات المعدودة حيث القيم السابقة ومفردات هذا النوع من البيانات تكون مرتبة ترتيباً تصاعدياً وصيغتها العامة هي كما يلي :

TYPE name=First data..last data ;

حيث تمثل First data المفردة الأولى من مجموعة البيانات الجزئية بينما تمثل Last data المفردة الأخيرة على أن يفصل بينها نقطتان متجاورتان وبحيث تقع بين المفردة الأولى والأخيرة مجموعة البيانات الجزئية المطلوبة بما فيها المفردتان الأولى والأخيرة .

مثال (2) : لبيان العطلة الصيفية يمكن ان نعر عنها بما يلي :

TYPE months=(January, February, March, April, May, June, July, August, September, October, November, December);

Holidays = June..August ;

حيث بينا في المثال أعلاه كيفية استعمال البيانات الجزئية تحت اسم العطلة Holidays والتي تقع مفرداتها بين شهري تموز وآب والتي تمثل العطلة الصيفية السنوية .

ويمكن ان تستخدم أيضا البيانات الصحيحة كبيانات جزئية

Months = 1.....12 ;

أو يمكن كذلك أن تستخدم أيضا البيانات الرمزية CHAR كبيانات جزئية :

CLASS NAME = 'ALI' .. 'AHMAD' ;

وكما أوضحنا سابقا فإنه يمكن استخدام بعض جمل الشرط والتكرار مع هذه البيانات مثل جملة FOR وجملة CASE .

مثال (3) : استخدام جملة CASE مع المعطيات الجماعية التعددية المعرفة من قبل المستخدم تكون على الشكل التالي :

CASE Months OF

February : WRITELN ('Start Semester');

March : WRITELN ('First Exam') ;

April : WRITELN ('Second Exam') ;

May : WRITELN ('Final Exam');

END;

(2-10) المصفوفات ARRAYS

(1-2-10) مفهوم المصفوفة في لغات البرمجة

عندما تستخدم متغيراً بسيطاً في أحد البرامج ، كما رأينا سابقاً ، يقوم الحاسوب بتحديد موقعاً تخزينياً واحداً يخزن فيه قيمة هذا المتغير ، فمثلاً إذا كان قيمة المتغير A تساوي 78 (عدد صحيح) ، عندئذ في كل مرة يظهر المتغير A أثناء تنفيذ البرنامج يقوم الحاسوب باستخدام القيمة المخزنة لـ A والتي هي 78 .

إذا كان A متغير يعبر عن درجات 20 طالب في مادة ما ، فإنه في مثل هذه الحالة نحتاج الى 20 موقعاً تخزينياً وكل موقع يحتاج الى اسم متغير مختلف ، لتمييز القيم التخزينية عن بعضها البعض ، وللتغلب على مشكلة استخدام اسماء متغيرات مختلفة لتسمية المواقع التخزينية ، يستخدم اسم مشترك لكل المواقع ويشار الى كل موقع برقم الذي يدعى دليل او مؤشر Index يدل على موقع العنصر ، وهذا يقابل في الرياضيات مصفوفة احادية البعد ويرمز لها بـ A_i حيث $i = 1, 2, \dots, 20$

ويمكن تعميم ما سبق على المتغيرات لبيانات متشابهة في النوع والمختلفة في القيم لأكثر من صفة ، فمثلاً نريد التعبير عن درجات 20 طالب في مادة ما لثلاثة شعب ، عندئذ نرمز بـ A_{ij} حيث $i = 1, 2, 3$ رقم الشعب و $j = 1, 2, \dots, 20$ رقم الطالب ، وهذا ما يقابل في الرياضيات مصفوفة ثنائية البعد ، وبالتالي عندما نكتب $A_{24} = 70$ نقصد بها ان درجة الطالب الرابع من الشعبة الثانية تعادل 70 ، ويمكن تشبيه عملية التخزين بالمصفوفة التالية :

~	1	2	3	4	...	20
1	80	30	50	33	...	60
2	83	45	80	70	...	55
3	72	90	74	65	...	84

والغاية مما سبق هو التعامل مع البيانات المتشابهة بالنوع من حيث نوعية البيانات (حقيقية، صحيحة، رمزية) ومن حيث ما يقصد بها واقعياً مثل (درجات عدد من

الطلبة في شعب مختلفة وفي جامعات مختلفة) ، والمختلفة بالقيم بشكل سهل ويسر ويعطي البرنامج مظهرا بسيطا غير معقدا .

(10-2-2) تمثيل عناصر المصفوفة في لغة باسكال

1) المصفوفة ذات البعد الواحد وتأخذ الصيغة التالية :

Array Name [Index] ;

حيث Array Name اسم المتغير والذي يجب التصريح عنه في بداية البرنامج.

Index دليل والذي يمكن ان يكون ثابت عددي او متغير عددي او تعبير حسابي .

والدليل يجب ان لا تتجاوز قيمته الحد الأقصى المحدد في صيغة الاعلان

مثال (4) :

- 1) GRADE [1] := 50 ;
- 2) GRADE [2] := 70 ;
- 3) SUM:=SUM+A[i] ;
- 4) READ (A[i]);

2) المصفوفة المتعددة الأبعاد وتأخذ الصيغة التالية :

Array Name [Index1, Index2] ;

حيث Array Name اسم المتغير والذي يجب التصريح عنه في بداية البرنامج .

حيث تمثل أدلة والتي يمكن أن تكون ثابت عددي أو متغير عددي	{	Index 1
		Index 2
أو		:
تعبير حسابي ، ويجب أن لا تتجاوز هذه الأدلة الحد الأقصى		:
المحدد في صيغة الإعلان .		:

مثال (5) :

- 1) GRADE [1,3] := 70 ;
- 2) X [5,20,4] := 80 ;
- 3) SUM = SUM + [i,j] ;
- 4) WRITELN (A [i,j]);

(10-2-3) الاعلان (التصريح) عن المصفوفات .

يتم الاعلان عن المصفوفات بنفس طريقة الاعلان عن المتغيرات مع إضافة الإعلان عن أبعاد المصفوفة .

أ) الصيغة العامة للإعلان عن مصفوفة أحادية البعد

VAR

Array Name : ARRAY [N1...N2] OF Type ;

حيث Array Name اسم المصفوفة .

N1...N2 مدى العناصر في المصفوفة أي ان N1 تعبر عن رقم

العنصر الاول من المصفوفة ، N2 العنصر الاخير من

المصفوفة .

نوع عناصر المصفوفات

Type

مثال (6) :

VAR

GRADE : ARRAY [1..20] OF INTEGER ;

نلاحظ ان عناصر المصفوفة من النوع الصحيح وان مدى الدليل ستكون من 1 وحتى 20 .

ب) الصيغة العامة للإعلان عن مصفوفة ثنائية البعد :

VAR

Array Name : [n1..n2,m1..m2] OF Type ;

حيث Array Name اسم المصفوفة

n1..n2 مدى العناصر للبعد الأول (عناصر الصفوف في

المصفوفة)

m1..m2 مدى العناصر للبعد الثاني (عناصر أعمدة المصفوفة)

مثال (7) :

```
VAR
  Grade : Array [ 1..3,1..20] OF REAL ;
```

نلاحظ أن البعد الأول هو من 1 وحتى 3 أي ثلاثة صفوف و البعد الثاني هو من 1 وحتى 20 أي عشرون عمودا.
(ج) الصيغة العامة للإعلان عن مصفوفة متعددة الأبعاد :

```
VAR
  Array Name : [n1..n2,m1..m2,L1..L2,...] OF Type ;
```

(10-2-4) أمثلة على استخدام ومعالجة المصفوفات :

مثال (8) : البرنامج التالي لحساب معدل درجات الطلبة في مادة المقدمة

لنرمز بـ A_i علامة الطالب i

N لعدد الطلبة

AVERAGE لتجميع علامات الطالب

i دليل كعداد

```
PROGRAM EXAMP (INPUT,OUTPUT) ;
```

```
VAR
```

```
  A : ARRAY [ 1..50] OF INTEGER ;
```

```
  N,I : INTEGER ;
```

```
  SUM, AVERAGE : REAL ;
```

```
BEGIN
```

```
  WRITE ('Enter Number Students :') ;
```

```
  READLN (N) ;
```

```
  FOR I:=1 TO N DO BEGIN
```

```
    WRITE ('A[', I,]=');
```

```
    READLN (A[I]) ;
```

```
  END;
```

```
  SUM:=0.0 ;
```

```
  FOR I :=1 TO N DO
```

```

SUM:=SUM+A[I] ;
AVERAGE:=SUM/N ;
WRITELN ('AVERAGE=' , AVERAGE:5:2) ;
END.

```

وعند التنفيذ ستبدو شاشة المخرجات كما يلي :

```

Enter Number Student :5
A[1]=65
A[2]=73
A[3]=55
A[4]=80
A[5]=97
AVERAGE=74.00

```

مثال (9) : اكتب برنامج بلغة باسكال لإدخال خمسة أعداد ثم ترتيبهم تنازلياً وطباعة العناصر المدخلة ، والعناصر المرتبة .

```

PROGRAM EXAM70 (INPUT,OUTPUT) ;
VAR
    X : ARRAY [1..50] OF INTEGER ;
    I,N,T : INTEGER ;
BEGIN
    WRITE ('Enter the Value of N=') ;
    READLN (N) ;
    FOR I :=1 TO N DO
        BEGIN
            WRITE ('X[' ,I ,']=') ;
            READLN (X[I]) ;
        END;
    WRITELN ('Unsorted Array is : ') ;
    FOR I := 1 TO N DO
        WRITE (X[I], ' ') ;
    FOR I :=1 TO N-1 DO
        BEGIN
            FOR K:= I+1 TO N DO
                BEGIN

```

```

        IF X[I]<=X[K] THEN
        BEGIN
            T:=X[I] ;
            X[I]:=X[K] ;
            X[K]:=T ;
        END ;
    END;
END;
WRITELN ('Sorted Array is :');
FOR I:=1 TO N DO WRITE (X[I],' ');
END.

```

وعند التنفيذ ستبدو شاشة المخرجات كما يلي :

Enter the Value of N=5
58
70
34
55
91
Unsorted Array is :
58 70 34 55 91
Sorted Array is :
91 70 58 55 34

مثال (10) اكتب برنامج لإدخال عناصر مصفوفة تتكون من خمسة صفوف وستة أعمدة ثم حساب مجموع عناصر كل صف .

```

PROGRAM EXAMEL (INPUT,OUTPUT);
CONST
    ROWSIZE = 5 ; COLSIZE=6 ;
VAR
    X: ARRAY [1..Rowsize,1..Colsize] OF INTEGER ;
    SUMROW : ARRAY [1..ROWSIZE] ;
    I,J : INTEGER ;
BEGIN
    FOR I:=1 TO 5 DO
        BEGIN

```

```

        FOR J:= 1 TO 6 DO
            READ (X[I,J]) ;
            WRITELN ;
        END;
    FOR I := 1 TO 5 DO
        BEGIN
            SUMROW [I] := 0.0
            FOR J := 1 TO 6 DO
                SUMROW [I] = SUMROW [I] + X[I,J] ;
            END;
            WRITELN ('SUM OF EACH ROW :') ;
            FOR I := 1 TO 5 DO
                WRITELN ('SUM OF ROW (' ,I,') = ',SUMROW [I]);
            END.

```

مثال (11) : اكتب برنامج بلغة باسكال لقراءة عناصر مصفوفة A ثم جد حاصل ضرب هذه المصفوفة بثابت عددي S علما بأن عدد صفوفها 7 وعدد أعمدتها هو 6 ثم بعد ذلك خزن عناصر المصفوفة الجديدة الناتجة من عملية الضرب في مصفوفة جديدة B

```

PROGRAM EXAMPEL (INPUT,OUTPUT) ;
VAR A : ARRAY [1..7,1..6] OF REAL ;
    B : ARRAY [1..7,1..6] OF REAL ;
    N : M : INTEGER ;
    S : REAL ;
BEGIN
    READLN (S) ;
    FOR N := 1 TO 7 DO
        FOR M := 1 TO 6 DO
            BEGIN
                READLN A[N,M] ;
                B [N,M] := S* A[N,M]
            END;
        FOR N:= 1 TO 7 DO
            BEGIN

```

```

FOR M:= 1 TO 6 DO
WRITE (B [N,M] ) ;
Writeln ;
END;
END .

```

(3-10) البرامج الفرعية SubPrograms

إن استخدام أسلوب التصميم الهرمي للبرمجة هو الأسلوب الناجح والكفاء في لغة باسكال حيث يوفر هذا الأسلوب سهولة في التصميم وسهولة في كتابة البرامج مع إمكانية التأكد من صحة البرنامج الذي تترابط أجزائه بصورة منطقية صحيحة حيث يتم باستخدام أسلوب TOP-DOWN والذي يشير إلى تقسيم البرنامج إلى أجزاء رئيسية بحيث يؤدي كل جزء وظيفة معينة في البرنامج ، ويربط هذه الأجزاء فيما بينها يمكن الوصول إلى برنامج متكامل ، وكذلك يمكن الرجوع إلى أي جزء من البرنامج في أي وقت . هذه الأجزاء من البرنامج المترابطة تسمى في لغة البرمجة (البرامج الفرعية) (Sub program) ، وفي لغة Pascal تسمى هذه البرامج الفرعية بـ (Procedure) الإجراء أو (Function) الاقتران.

هذه البرامج الفرعية تكون سهلة التصميم حيث يقوم المبرمج بإعدادها بشكل منفصل ثم يتم تجميعها وربطها بالبرنامج العام ، ولكون هذه البرامج الفرعية صغيرة ومستقلة وغير متشعبة فإنها تمكن المبرمج من التأكد من صحتها وصحة نتائجها بسهولة وللبرامج الفرعية فائدة كبيرة وهامة في المسائل المعقدة خاصة والتي تحتوي على التكرار المتعدد لبعض جمل باسكال ، حيث يتم التعويض عن التكرار المتعدد بأسلوب استدعاء الاجراء او الاقترانات عند الحاجة وبهذا يتم اختصار حجم البرنامج الى حد كبير . وسوف نقوم بشرح وتفصيل كل من الإجراءات والاقترانات ومعرفة استخدامها والفرق فيما بينها .

(10-3-1) الإجراءات Procedures

في لغة باسكال Pascal هناك نوعان من الإجراءات :

الأول : هو الإجراء الداخلي Internal وهو الذي يتم إعداده داخل البرنامج الرئيسي ويكون موقعه بعد تعريف جميع الاقتارات في البرنامج .

الثاني : هو الإجراء الخارجي External وهو الذي يتم إعداده خارج البرنامج الرئيسي وبشكل مستقل ومنفصل تماما ويتم استخدامه بوضع اسمه مع المتغيرات المستخدمة في تعريف الإجراء داخل البرنامج الرئيسي .

إن الإجراءات تستخدم عند الحاجة لحساب قيمة لأكثر من عنصر واحد عند استدعاء البرنامج الفرعي لاستخدامها في البرنامج الرئيسي حيث يبدأ تعريف الإجراء باستخدام كلمة Procedure ثم يليها اسم الإجراء وبعدها يتم وضع قائمة المتغيرات التي يعتمد عليها الإجراء وبحيث يكون ترتيب وتنظيم الإجراء مثل تنظيم البرنامج الاعتيادي والصيغة العامة للإجراء Procedure هو كما يلي :

PROCEDURE name (Parameter List);	
Local Variable declaration section	
BEGIN	
-----	} جمل باسكال تمثل جسم الإجراء

END.	

حيث :

PROCEDURE عنوان الإجراء وهي كلمة محجوزة

name اسم الإجراء الذي يتبع الكلمة المحجوزة PROCEDURE

Parameter List تمثل قائمة المتغيرات أو بارامترات الإجراء والتي يتم تعويضها في جملة الاستدعاء في البرنامج الرئيسي بالمتغيرات الفعلية التي يجب ان تتطابق مع المتغيرات المحايدة من حيث النوع والعدد .

Local Variable declaration section مقطع الاعلان عن المتغيرات التي ستستخدم داخل الاجراء فقط وتدعى متغيرات موضعية ، بينما المتغيرات التي يصرح عنها في البرنامج الرئيسي فتدعى المتغيرات الشاملة والتي يكن استخدامها في مكان من البرنامج الرئيسي والبرامج الفرعية .

و يجب أن يتبع كل بارامتر في **Parameter List** نوعه ، وتقسم عادة هذه البارامترات الى نوعين :

- متغيرات مدخلات : ويتم تعريف قيمتها من خلال المتغيرات الفعلية الواردة في جملة الاستدعاء في البرنامج الرئيسي والتي تقابلها في الترتيب وتتطابق معها في النوع .

- متغيرات مخرجات : ويتم حسابها داخل الإجراء وتسند قيمتها إلى المتغير الفعلي في جملة الاستدعاء في البرنامج الرئيسي .

- وعادة نرتب قائمة متغيرات الإجراء المدخلات أولا ومن ثم الكلمة المحجوزة **VAR** وبعدها متغيرات المخرجات .

مثال (12) أكتب برنامج بلغة باسكال لاختيار العدد الأكبر من بين ثلاثة أعداد

معلومة مستخدما **PROCEDURE**

```
PROCEDURE MAXIMUM (A,B,C : INTEGER) ;
```

```
VAR
```

```
    MAX : INTEGER ;
```

```
BEGIN
```

```
    IF A>B THEN MAX:=A ELSE MAX:=B ;
```

```
    IF C>MAX THEN MAX:=C ;
```

```
    WRITELN('THE MAXIMUM =' ,MAX)
```

```
END;
```

مثال (13) اكتب برنامج بلغة باسكال (إجرائي) لتبديل قيم متغيرين فيما بينهما.

نلاحظ في هذا السؤال انه يجب علينا أن نستخدم متغير وسيط بين المتغيرين أعلاه لكي نقوم بتحويل قيم أحدهما للآخر ولهذا وجب عند استخدام متغير وسيط تعريفه وتحديد نوعه في مقطع التصريح داخل برنامج الإجراء

```
PROCEDURE SWAP (X,Y : REAL) ;  
VAR  
    TEMP : REAL ;  
BEGIN  
    TEMP :=X; X:=Y; Y:=TEMP ;  
END;
```

مثال (14) اكتب برنامج بلغة باسكال لإيجاد أكبر عدد وأصغر عدد من مجموعة أعداد باستخدام الإجراء .

```
PROGRAM F_MAX_MIN(INPUT,OUTPUT) ;  
VAR  
    N :INTEGER ;  
PROCEDURE MAX_MIN (NUM : INTEGER);  
    MAX,MIN,X : REAL ;  
    I : INTEGER ;  
BEGIN  
    WRITELN ('ENTER',NUM,'NUMBER PLEASE ..');  
    READ (X) ;  
    MAX :=X ;  
    MIN := X ;  
    FOR I := 1 TO NUM-1 DO  
        BEGIN  
            READ (X) ;  
            IF X >MAX  
                THEN  
                    MAX :=X ;  
            IF (X<MIN)  
                THEN
```

```

MIN := X ;
END;
WRITELN('MAX IS:',MAX :4:2,',MIN IS:',MIN:4:2);
END;
BEGIN
    CLRSCR ;
    WRITELN('PLEASE ENTER HOW NUMBER YOU WANT:');
    READ (N) ;
    MAX MIN (N) ;
    REPEAT UNTIL REYPAESSED ;
END.

```

مثال (15) أكتب برنامج بلغة باسكال لحساب مربع عدد حقيقي X لجميع القيم المحصورة بين 0 و 1 بزيادة مقدارها 0.1 باستخدام الاجراء .

```

PROGRAM EXAMPEL (INPUT,OUTPUT) ;
VAR
    X,Y : REAL ;
PROCEDURE Sqrnumber (X : REAL ; VAR X2 :REAL);
    BEGIN
        X2 := X * X ;
    END;
BEGIN (* MAIN PROGRAM *)
    X:=0.0 ;
    WHILE X < 1.0 DO BEGIN
        Sqrnumber (X,Y);
        WRITELN ('SQR('X,')='Y:S:2) ;
        X:=X+0.1 ; END;
END.

```

(10-3-2) الاقترانات FUNCTIONS

إن الاقتران في لغة باسكال يستخدم عادة لحساب قيمة معينة واحدة بينما يتم استخدام الاجراء لحساب مجموعة من القيم ، حيث يتم تعريف الاقتران عادة في بداية البرنامج بعد تعريف الجزء الخاص بوحدة التخزين اللازمة للبرنامج، على ان يعامل اسم الاقتران كاسم متغير في البرنامج ، ولهذا فإن تنفيذ

الاقتران في البرنامج يتم فوراً بمجرد ذكر اسم الاقتران ، لهذا فمن الواجب ذكر نوع الاقتران هل هو حقيقي ، ام صحيح ، ام رمزي ، ام منطقي .
علما ان هناك نوعان من الاقترانات وهما :

1. الاقترانات المكتبية Library Functions

2. اقترانات المبرمج التعريفية Internal defined functions

و الآن سوف نقوم بشرح هذين النوعين من الاقترانات :

(1-2-3-10) الاقترانات المكتبية Library Functions

هذه الاقترانات تكون معرفة في لغة باسكال أصلا وهي مجموعة من الاقترانات القياسية الجاهزة للاستخدام بمجرد ذكر اسمها وفي أي موقع من مواقع البرنامج ولا يوجد هناك ضرورة لتعريفهما في البرنامج من هذه الاقترانات على سبيل المثال :

SQRT (X)	SQR (X)	ABS (X)
LN (X)	COS (X)	SIW (X)
ROUND (X)	TRUNC (X)	EXP (X)
CHR (X)	ORD (X)	ARCTAN (X)
ODD (X)	SUCC (X)	PRED (X)

من الضروري في هذه الاقترانات مراعاة ان تكون المتغيرات الفعيلة المستخدمة مطابقة فعلا لنوع المتغيرات المحايدة المعرفة لكل اقتران قياسي وقد تناولنا هذه الاقترانات في الفقرة (9-2-8) .

(1-2-3-10) اقترانات المبرمج التعريفية Internal defined Functions

هذه الاقترانات يتم كتابتها وتعريفها داخل البرنامج وفي بدايته ومن ثم استخدامه داخل الجزء التنفيذي من البرنامج ، ولهذا يسمى هذا النوع من الاقترانات ايضا بالاقترانات الداخلية وصيغته العامة :

```

FUNCTION name (a1,a2,a3 ... an : type ;
    Local Variable declaration section
BEGIN
    -----
    -----
    -----
    -----
END;

```

}
 }
 }
 }

جمل باسكال
 تشكّل جسم الاجراء

مثال (16) : اكتب برنامج اقتران لإيجاد معدل قيمتين.

```

FUNCTION AVG (x,y :REAL) : REAL ;
BEGIN
    AVR:=(X + Y) / 2.0 ;
END;

```

مثال (17) : اكتب برنامج اقتران لإيجاد مساحة دائرة معلوم نصف قطرها .

```

FUNCTION AREA (Y:REAL):REAL ;
BEGIN
    AREA :=3.14159*Y*Y ;
END;

```

مثال (18) : اكتب برنامج اقتران لحساب القيمة العظمى لقيمتين معلومتين

```

FUNCTION LARGER (D,E : REAL) : REAL ;
BEGIN
    IF D>E THEN
        LARGER :=D
    ELSE
        LARGER :=E ;
END;

```

مثال (19) اكتب برنامج اقتران لحساب القيمة المطلقة |S| و المعرفة على النحو

التالي :

$$Y(S) = \begin{cases} S & \text{if } S > 0 \\ -S & \text{if } S < 0 \\ 0 & \text{if } S = 0 \end{cases}$$

```

FUNCTION
    ABSLT (S:REAL):REAL ;
BEGIN
    IF S<0 THEN ABSLT:=-S ELSE
    IF S=0 THEN ABSLT:=0 ELSE
    ABSLT:=S ;
END;

```

مثال (20) : اكتب برنامج بلغة باسكال لحساب معدل درجات 30 طالب
 باستخدام الإجراء لحساب المعدل .

```

PROGRAM EXAM17 (INPUT,OUTPUT) ;
TYPE
    Vector =ARRAY[1..30]OF REAL ;
VAR
    GRADE :Vector ;
    AVG    : REAL ;
    I      : INTEGER ;
PROCEDURE Avereg (N:Integer; Y:Vector; VAR
Z:REAL);
VAR
    SUM : REAL ;
BEGIN
    SUM :=0.0 ;
    FOR I:=1 TO N DO
    SUM:=SUM + Y[I] ;
    Z:=SUM/N ;
END;
BEGIN
    FOR I:=1 TO 30 DO READ (GRADE[I]);
    Avereg (5, GRADE , AVG);
    WRITELN (AVG:4:2);
END.

```

مثال (21) اكتب برنامج لحساب مضروب عدد باستخدام الاقتران حيث :

$$N! = \begin{cases} 1 & ; n=0 \\ 1*2*3*....*n & ; n \neq 0 \end{cases}$$

```
PROGRAM FACTRIAL (INPUT,OUTPUT);
VAR
    N : INTEGER ;
FUNCTION FAC (K : INTEGER) : INTEGER ;
VAR
    F,I : INTEGER ;
BEGIN
    F :=1 ;
    FOR I := 1 TO K DO
        F := F*I ;
    FAC := F ;
END ;
BEGIN
    CLRSCR ;
    WRITE ('N=') ; READLN (N) ;
    WRITE (N,'!=', FAC(N)) ;
    READLN ;
END.
```

مثال (22) اكتب برنامج بلغة باسكال لطباعة قيم كل من X ، X^3 ، لجميع قيم

X من 1 الى 5 بزيادة مقدارها 0.5 مستخدما الاقتران لحساب X^3

```
PROGRAM EXAMPEL (INPUT,OUTPUT) ;
VAR
    X,Y : REAL ;
FUNCTION CUBE (A:REAL) : REAL ;
BEGIN
    CUBE:=A*A*A ;
END;
BEGIN (* MAIN PROGRAM *)
```



```

X:= 1.0 ;
WRITELN ('X      Y');
WRITELN ('---    ---');
WHILE X <=5.0 DO
    BEGIN
        Y:=CUBE (X)
        WRITELN (X:5:2 , '    ' , Y:6:2);
    END;

```

END.

RECORDS السجلات (4-10)

ان السجلات هي من المعطيات البنائية (Structured Types) في لغة باسكال وتعتبر ذات مقدرة عالية في معالجة البيانات .

لقد تطرقنا فيما سبق الى معالجة معلومات من نفس النوع (رقمية) أو حرفية أو رمزية أو مصفوفات ولكن في بعض الأحيان نحتاج لمعالجة معلومات مختلفة النوع، ففي المصفوفات مثلاً كنا نعالج منظومات مرتبطة فيما بينها مثل درجات الطلاب وأسماء الطلاب وكانت الطريقة معقدة في بعض الأحيان ، لذلك فمن الأفضل التعامل مع السجلات التي تمكننا من معالجة معلومات من أنواع مختلفة ، حيث أن المعلومات المختلفة في السجل تحتل مجالات خاصة بكل نوع تدعى حقول فمثلاً في حالة مجموعة من الطلبة وأرقامهم وموادهم وامتحاناتهم نجد أن اسم الطالب يمثل حقلاً للأسماء ورقم الطالب يمثل حقلاً للأرقام ومادة الطالب تمثل حقلاً للمواد ودرجات الطالب تمثل حقلاً للمعلومات وهكذا .

فلو فرضنا أن السجل الذي يحوي أسماء وأرقام ودرجات ومواد الطلبة

يكون بالشكل التالي :

اسم الطالب name	رقم الطالب No.	المادة Course	الامتحان الاول First Ex	الامتحان الثاني Second Ex	الامتحان النهائي Final Ex	المجموع sum

فهذا الجدول يمثل حقول ولكل حقول خصوصية خاصة في السجل ولهذا نستطيع

تعريف السجل من خلال الجدول أعلاه بإحدى الطريقتين :

الطريقة الأولى : يتم تعريف السجل وإدخاله كجزء من الإعلان عن المتغيرات (Variables) وهذا التعريف يتم الاستفادة منه في تعريف أنواع متغيرة من السجلات المستقلة تماما كما يتم تعريف المتغيرات البسيطة والمنظومات المتغيرة ولهذا فإن الصورة العامة لتعريف السجل في هذه الحالة يكون على الشكل التالي :

VAR name : RECORD

field 1 ;

field 2 ;

field 3 ;

:

field n ;

END.

وفي مثالنا السابق فإن field 1 يمثل الاسم
field 2 يمثل الرقم
field 3 يمثل المادة
field 4 يمثل العلامة
field 5 يمثل المجموع
و name تمثل اسم السجل

وهذه جميعها تمثل الإعلان
عن المجالات المختلفة الداخلة
في السجل

علما بأن المجالات المختلفة الداخلية في السجل تكتب بطريقة تماثل طريقة الإعلان عن المتغيرات المستقلة أي :

Field name : type ;

حيث Filed Name اسم المجال .

TYPE نوع المجال .

مع ملاحظة وجود الفارزة المنقوطة التي تفصل بين المجالات المستقلة ضمن السجل الواحد

مثال (23) لدينا سجل طالب يتكون من رقم الطالب ورمز معدله التراكمي ودرجته في مادة معينة ، يمكن تمثيل ذلك في متغير سجلي يسمى Student وعلى أساس أن الرقم يتألف من (أرقام صحيحة) والمعدل التراكمي يتألف من (حروف) ودرجة الطالب تتألف من (أعداد عشرية) ولهذا يمكن كتابة السجل كما يلي :

VAR

```
STUDENT      : RECORD
STUDNO       : INTEGER ;
POINT        : CHAR ;
TOTGRADE     : REAL ;
END ;
```

نلاحظ ان هذا السجل يحوي على ثلاث مجالات المجال الأول سمي STUDNO رقم الطالب ، المجال الثاني سمي POINT والذي احتوى رمز معدل الطالب التراكمي في المادة فمثلا بالحروف CHAR والمجال الثالث سمي TOTGRADE والذي يحوي درجة الطالب في المقرر المعين ومن الضروري ان تكون مكونات السجل من الأنواع المعروفة والتي يتم تعريفها في البرنامج وقبل تعريف السجل .

الطريقة الثانية : يتم تعريف السجل على أساس النوع (TYPE) وليس كمتغير ثم بعدها يتم الإعلان عن المتغيرات السجلية وهذا التعريف متطابق مع تعريف المصفوفات (Arrays) ، ولهذا فهي أكثر انتشارا من الطريقة الأولى حيث إننا نستطيع تعريف أكثر من متغير سجلي من النوع الذي تم تعريفه وهنا يتم الاستعاضة عن كلمة VAR بكلمة TYPE والصيغة العامة لهذه الحالة :

<pre>TYPE name = RECORD field 1; field 2; : field n; END</pre>

حيث NAME اسم السجل

تمثل الإعلان عن المجالات المختلفة الداخلية في السجل

{	field 1
	field 2

	field n

مثال (24) : لدينا سجل طالب يتكون من رقم الطالب ورمز معدله التراكمي ودرجته في مادة معينة ، يمكن تمثيل ذلك في متغير سجلي يسمى STUDENT وعلى أساس أن الرقم يتألف من (أرقام صحيحة)، والمعدل التراكمي يتألف من حروف، ودرجة الطالب تتألف من أعداد عشرية، لذلك يتم تعريف نوع سجلي اسمه GRADE ومتغير سجلي اسمه STUDENT يكون من نوع GRADE وكما يلي :

```
TYPE      GRADE : RECORD
          STUDENT : INTEGER ;
          POINT : CHAR ;
          TOTGRADE : REAL ;
          END;
VAR       STUDENT : GRADE ;
```

(5-10) الملفات Files

الملف هو عبارة عن مجموعة من السجلات التي تضم عدد من الحقول المختلفة النوع والطول ، ينتهي كل ملف في لغة باسكال بعلاقة نهاية الملف EOF أي (End Of File) وتتصف جميع الملفات بلغة باسكال بأنها ملفات تتابعية أي ان النفاذ الى الملف يتم بالتسلسل أي عنصر تلو الآخر . ويتم الإعلان (التصريح) عن الملف بالصيغة العامة التالية :

TYPE file name = FILE OF Type ;

حيث file name اسم الملف
Type نوع الملف والذي يمكن ان تكون , REAL , CHAR , INTEGER , TEXT
ويمكن التمييز بين نوعين من الملفات :

1. ملفات الإدخال Input Files : وهي ملفات معدة للقراءة من قبل البرنامج الرئيسي .

2. ملفات الإخراج Output Files : وهي ملفات معدة لتخزين وحفظ البيانات حيث يتم تسجيل البيانات عليها ليتم استدعائها في السبرامج وقت الحاجة .

وللتعامل مع الملفات ومعالجتها يستخدم جمل مختلفة أهمها :

(1) REWRITE (F) لفتح أو إنشاء ملف جديد فارغ باسم F .
(2) WRITE (F,X) لكتابة القيمة X على الملف F .
(3) ASSIGN(F,'file name') تنسيب اسم الملف F إلى ملف خارجي باسم file name .

(4) CLOS (F) إغلاق ملف مفتوح .

(5) RENAME يستخدم لتغيير اسم ملف .

(6) RESET (F) يفتح ملف موجود مسبقاً وذلك للقراءة

(7) READ (F,X) لقراءة القيمة X من الملف F .

(8) EOF إشارة إلى نهاية الملف .

(9) EOLN إشارة إلى نهاية السطر .

بالإضافة إلى جمل أخرى .

مثال (25) : البرنامج التالي لإنشاء ملف يحفظ نتائج البرنامج الذي يحسب مربع

عدد

```
PROGRAMME EXAMPEL1(INPUT,OUTPUT,FILES) ;  
TYPE SQRFILE = FILE OF INTEGER ;  
VAR  
    X: SQRFILE ;  
    I,J,N : INTEGER ;  
BEGIN  
    CLRSCR ;
```

```

WRITELN ('ENTER N = ' );
READLN (N) ;
ASSIGN (X,'A: TEST . DAT ' ) ;
REWRITE (X) ;
I:=1 ;
WHILE I <= N DO
    BEGIN
        J:=SQR(I) ;
        WRITELN (I,' ',J) ;
        WRITE (X,J) ;
        I :=I+1 ;
    END;
CLOSE (X) ;
END.

```

من المثال السابق نلاحظ أننا صرحنا عن الملف على انه من نوع الصحيح في بداية البرنامج وباستخدام الجملة ASSIGN اسندنا الملف X إلى الملف TEST.DAT على القرص المرن الموجود على السواقة A ، ثم ضمن حلقة التكرار المشروط WHILE حسبنا مربعات الأعداد وضمن جملة WRITE (X,J) نخزننا هذه المربعات J على الملف X ثم أغلقنا الملف المفتوح باستخدام الجملة CLOSE .

مثال (26) : البرنامج التالي لقراءة الملف الذي أنشأناه في المثال السابق ثم جمع تلك القيم وعرضها على الشاشة :

```

PROGRAM EXAMPEL2(INPUT,OUTPUT,FILES) ;
TYPE SUMFILE = FILE OF INTEGER ;
VAR
    X: SUMFILE ;
    I,SUM : INTEGER ;
BEGIN
    CLRSCR ;
    ASSIGN (X,'A: TEST , DAT ' ) ;
    RESET (X) ;

```

```

SUM:=0 ;
WHIL NOT EOF (X) DO
    BEGIN
        READ (X,I) ;
        WRITELN ('I=',I) ;
        SUM :=SUM+I ;
    END;
CLOSE (X) ;
WRITELN ('THE TOTAL VALUES = ',SUM:6) ;
END.

```

من المثال السابق نلاحظ أننا باستخدام الجملة RESET فتحنا الملف X المخزن مسبقاً ثم باستخدام الجملة READ(X,I) قرأنا القيمة I من الملف X ثم جمعنا هذه القيم (التي تمثل مربعات الأعداد) أظهرنا النتائج على الشاشة بعد أن أغلقنا الملف المفتوح باستخدام الجملة CLOSE (X) .

مثال (27) : البرنامج التالي يتضمن إنشاء ملف لإدخال البيانات على شكل سجلات يتألف هذا السجل من رقم الزبون ، اسم الزبون ، رقم الهاتف .

```

PROGRAM EXAMPEL3(INPUT,OUTPUT,FILES) ;
TYPE COSTUMERFILE = FILE OF TEXT ;
VAR
    NEW : COSTUMERFILE ;
    NUMBER,NAME,TEL : STRING[20] ;
    N : INTEGER ;
BEGIN
    ASSIGN (NEW,'A: TEST . TXT ');
    REWRITE (NEW) ;
    N:=5 ;
    WHIL N>0 DO
        BEGIN
            WRITE ('Enter-1 If You Want End') ;
            READLN (N) ;
            WHIL NOT EOLN DO
                BEGIN

```

```
        WRITE ('Customer Number=');
        READLN (NUMBER);
        WRITE ('Customer Name =');
        READLN (NAME);
        WRITE ('Telephon=');
        READLN (TEL);
        WRITE('Enter-1If you want End
              the Record');
        READLN (N);
        WRITE(NEW,NUMBER,NAME,TEL);
        END;
        END;
    CLOSE (NEW) ;
END.
```


تمارين واسئلة عامة

1- إذا علمت أن :

TYPE Day Of Week =(SAT,SUN,MON,TUE,WED,THR,FRI);

فأوجد قيمة كل من :

1. PRED (MON)
2. ORD (SAT)
3. SUSS (TUE)
4. ORD (SUCC(THR))
5. ORD (PRED (MON))

2- اكتب برنامج بلغة باسكال لطباعة أسماء الأشهر الستة ورقم الشهر بالتسلسل لزاء كل واحد منهم مستخدما بيانات المستخدم التعريفية .

3- اكتب برنامج لإيجاد اكبر عنصر في مصفوفة ما مع دليله مستخدما بذلك الإجراءات .

4- اكتب برنامج لإيجاد اكبر واصغر عدد من مجموعة أعداد معطاه باستخدام الإجراءات .

5- اكتب برنامج بلغة باسكال لحساب لوغاريتم أي عدد باستخدام الاقتارات.

6- اكتب برنامج بلغة باسكال لإيجاد اكبر عنصر واصغر عنصر من عناصر مصفوفة A .

7- اكتب برنامج لإيجاد اكبر عنصر في كل سطر من عناصر مصفوفة ما .

8- اكتب برنامج لإيجاد اصغر عنصر في كل عامود من أعمدة مصفوفة ما .

9- اكتب برنامج لحل جملة معادلات باستخدام طريقة غوص جوردان .

10- اكتب برنامج ترتيب مجموعة من القيم المدخلة بشكل عشوائي ترتيبا تصاعديا.

- 11- اكتب برنامج لحساب درجات الطلاب المتقدمين لامتحان المقدمة ومقدار انحراف كل علامة عن المعدل ، مع العلم ان

$$\frac{\text{معدل} = \frac{\text{مجموع علامات الطلبة}}{\text{عدد الطلبة}}}$$

انحراف كل علامة عن المعدل = علامة الطالب المعدل .
على ان تبدو شاشة المخرجات بالشكل التالي :

AVERAGE =	
العلامة	الانحراف
<u>SCORE</u>	<u>DEVIATION</u>
-----	-----
-----	-----

- 12- اكتب برنامج بلغة باسكال يتضمن ملف إدخال بيانات على شكل سجلات حيث يتألف السجل من رقم الشخص ، اسم الشخص ، الشارع ، المدينة ، رقم الهاتف وينتهي إدخال السجلات عند إعطاء القيمة صفر .

الفصل الحادي عشر

أساسيات بيئة باسكال

(1-11) الدخول إلى بيئة باسكال PASCAL والخروج منها :

بعد تشغيل جهاز الحاسوب وظهور محث النظام في الزاوية العليا اليسرى من الشاشة كما يلي :

C:\>

ندخل الحرفين TP وهما اختصار لكلمتي Turbo Pascal ثم نضغط مفتاح الإدخال ، أي

C:\>TP

تظهر رسالة على الشاشة

Turbo Pascal version 7.0 copyright (c) 1983,92 Borland International .

وبعد هذه الرسالة تظهر نافذة الباسكال التي سنتوقف عند بعض تفاصيلها في الفقرة التالية .

للخروج من بيئة الباسكال والعودة إلى شاشة الدوس يمكن استخدام عدة أساليب إلا أن الطريقة السريعة للخروج من بيئة الباسكال أينما كنا فيها هو الضغط على مفتاحي ALT + X .

ملاحظة : نستخدم أحيانا للدخول إلى بيئة الباسكال كلمة TURBO أو TURBO PASCAL حسب اسم الملف التنفيذي المثبت على جهاز الحاسوب .

(11-2) أقسام النافذة الرئيسية لبيئة الباسكال (نافذة التنقيح

(Edition Window

النافذة الرئيسية أو نافذة التنقيح هي المكان الذي نكتب وتنقح فيه البرنامج المكتوب بلغة الباسكال بالاستعانة بنوافذ أخرى مساعدة ، ليصار إلى تصميمه وتحويله إلى برنامج منطقي وتشغيله وتوثيقه بالإضافة إلى خدمات أخرى .
أجزاء هذه النافذة من الأعلى إلى الأسفل كما هو موضح في الشكل (1) :

1- شريط القوائم أو Menu Bar

ويضم عشرة أسماء ، تشكل عناوين للقوائم الفرعية ، حيث كل قائمة تضم مجموعة من الأوامر التي تساعد المبرمج الوصول إلى المهام التي يريد إنجازها بطرق مختصرة وسهلة ، هذه القوائم هي :

File – Edit – Search – Run – Compile – Debug – Tools – Options – Window – help .

عند تشغيل TP 7 (تريبو باسكال 7) نلاحظ ان كل اسم قائمة يحتوي على حرف واحد بلون مختلف .

2- شريط العنوان Title Bare ويضم من اليسار إلى اليمين :

أ. علبة الاقفال [Close Box] :

وظيفتها تمكين المستخدم من إقفال النافذة التي يعمل عليها بسرعة باستخدام الفارة . وذلك بوضع مؤشر الفارة على العلبة ثم النقر على الزر اليسر وبالتالي يتم اقفال النافذة الحالية وعرض النافذة التالية (أي الملف التالي) .

ب. عنوان النافذة Title :

ويقع في وسط الشريط يحتوي على عنوان النافذة التي تعمل عليها ، أي اسم الملف الذي يحتوي البرنامج الموجود في النافذة .

عند البدء بتشغيل TP 7 فان نافذة العنوان تأخذ الاسم التالي NONAME00.PAS أي بلا عنوان ، ولعنوان النافذة فائدتين :

1) تعريف الملف مع الامتداد الموجود على النافذة .
 2) اذا كان قد فتحنا اكثر من نافذة في وقت واحد على الشاشة فيمكن الوصول إلى احدى النوافذ ووضعها في المقدمة ، وذلك بوضع مؤشر الفأرة على العنوان المراد تفعيله ثم نقر بالمفتاح الايسر نقرة مزدوجة .
 ج. رقم النافذة "1_"

هذا الرقم يزداد مع زيادة عدد النوافذ التي نفتحها ضمن نافذة الباسكال ، ويساعد على تفعيل النافذة التي نريدها وذلك عن طريق مفتاح ALT ورقم النافذة المطلوبة .

وبالبدل عن ذلك يمكن استخدام القائمة Window والأمر Next أو القائمة Window و الأمر List ويظهر عندئذ صندوق حوار يعرض أسماء جميع الملفات المفتوحة على نافذة الباسكال نحدد احداها ونضغط مفتاح الادخال Enter .

```
File Edit Search Run Compile Debug Tools Options Window Help
PRNFLT.R.PAS
end;

procedure ProcessCommandLine;
var
  Param: String;
  I: Integer;


  function ParamVal(var P: String; Default: Word): Word;
  var
    N, E: Integer;
  begin
    Delete(P, 1, 1);
    Val(P, N, E);
    if E = 0 then
      ParamVal := N
    else
      ParamVal := Default;
    end;
  end;


begin
  PrinterCodes := @AsciiCodes;
  206:1
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu
```

الشكل (1) نافذة باسكال

د. علبة التحكم بحجم النافذة [] Zoom Box

نستخدم لتكبير حجم النافذة لتملا الشاشة بكاملها أو إعادة حجم النافذة إلى وضعها الطبيعي وذلك حسب شكل ظهور العلبة :

* إذا كانت من الشكل [] هذا يعني انه يمكن تكبير حجم النافذة اكبر لتملا الشاشة بكاملها وذلك بالنقر على العلبة باستخدام الفارة .

* إذا كانت من الشكل [] يعني ان النافذة الفعالة في أقصى حجم لها ويمكن العودة إلى الحجم الطبيعي بالنقر على العلبة باستخدام الفارة .

وبالبدل عن استخدام الفارة هو القائمة Window والامر Zoom .

3- شريط التدرج العمودي Vertical Scroll

الموجودة في الجهة اليمنى من الشاشة ، ويعتبر شريط التدرج العمودي :

1. اداة لعرض الاجزاء المخفية من النافذة الفعالة باستخدام الفارة وذلك بالنقر على الاسهم ▼ أو ▲ .
- البديل عن الفارة استخدام مفاتيح الاسهم ▼ أو ▲ من لوحة المفاتيح .
2. مؤشر لتحديد الموقع المعروض بالنسبة للملف كاملا ، فإذا كانت علبة التدرج ☐ في منتصف شريط التدرج العمودي يعني أننا في منتصف الملف المفتوح على الشاشة .

ويمكن تغيير موقع علبة التدرج لعرض أجزاء أخرى من الملف وذلك باستخدام الفارة بالنقر على العلبة و السحب .

4- شريط التدرج الافقي Horizontal Scroll Bar

الموجود في اسفل الشاشة قبل الشريط الاخير ، ويعتبر هذا الشريط اداة لعرض الاجزاء المخفية من النافذة اذا كان الملف المعروض اكبر من استيعاب النافذة أفقيا وذلك اما بالنقر على الاسهم ► أو ◀ التي تتحرك باتجاه

اليمين أو اليسار . واما باستخدام الاسهم ← أو → من لوحة المفاتيح.

ملاحظة : اذا تقرنا بالفارة على المساحة المظللة الواقعة بين الاسهم في شريط التدرج تنتقل صفحة كاملة في الملف (أي نافذة كاملة Window Full).

5- موقع الاشارة "1.1" Cursor Location

هذا الموقع يبين احداثيات المشيرة Cursor (الروامض) على النافذة وتدعى المشيرة باسم مؤشر الادراج .

مثلا اذا كانت من الشكل 1.1 يعني ان المشيرة في السطر الأول والعمود الأول من النافذة أي انهما في الزاوية العليا اليسرى

اذا كان من الشكل 3.5 يعني ان المشيرة في السطر الثالث والعمود الخامس .

6 - النافذة الفعالة Active Window

وهي المنطقة الواقعة بين شريط العنوان وشريط التدرج الافقي والعمودي والتي يمكننا ادخال أوامر الباسكال وتنقيحها وتشغيل البرنامج باستخدام القوائم

- وتستوعب النافذة إلى 256 حرفا (رمزا) على الأكثر في الخط الافقي الواحد .

- مهما فتحنا نوافذ على الشاشة فإنه هناك نافذة واحدة فقط فعالة في وقت واحد .

- أي امر نختاره أو نص يطبق على النافذة الفعالة .

7 - شريط الرسائل Message Bar

يعطي هذا الشريط رسالة مختصرة جدا عن المهمة التي نريد القيام بها بعد انجاز الخطوات المناسبة بها .

عند بداية تشغيل TP 7 تحوي الرسالة المعلومات التالية :

F1 Help F2 Save F3 Open Alt-F9 Compil F9 Make
F10 Menu .

الا ان هذه الرسالة غير ثابتة فهي تتغير مع تغيير المهمة التي ننوي انجازها ،
الا ان كل ما يستطيع TP 7 القيام به في هذه اللحظة من مهمات هو
الانجازات الستة 6 المعروضة في الرسالة ومنها نبدأ بإنجاز مهمات اخرى
تدرجيا .

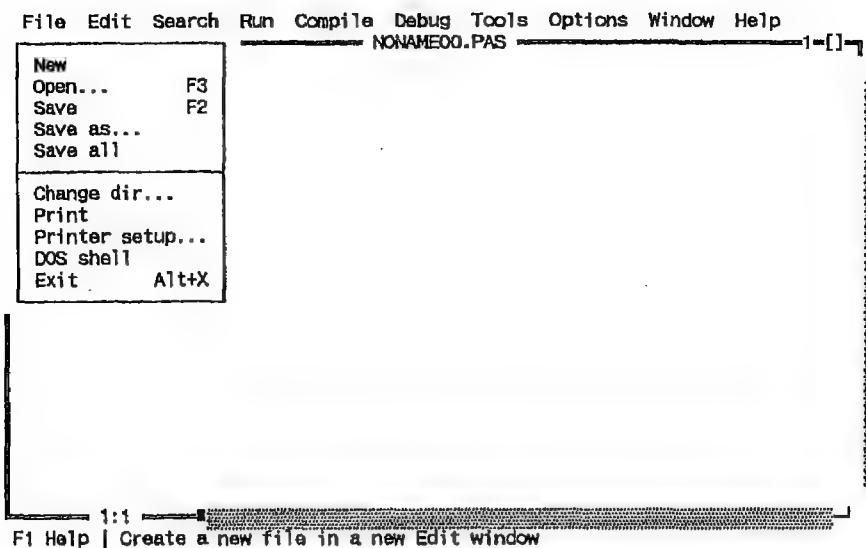
(3-11) تشغيل شريط القوائم :

لتشغيل شريط القوائم ، وفتح القوائم المتفرعة منه نتبع احدى الطرق التالية :
(1) نضغط على المفتاح F10 فيصبح شريط القوائم فعال ثم نستخدم مفاتيح
الاسهم ← أو → لتحديد اسم القائمة التي نريد فتحها ثم نضغط
مفتاح الإدخال Enter .
(2) نضغط المفتاح Alt مع الحرف المميز من اسم القائمة المراد فتحها ، مثلا عند
الضغط على مفتاح Alt مع الحرف F نفتح القائمة File.
(3) باستخدام الفارة ، نضع مؤشر الفارة على اسم القائمة المراد فتحها وننقر
بالزر الايسر فتفتح القائمة .
بعد فتح القائمة ، بغض النظر عن الطريقة المتبعة، نلاحظ ان كل قائمة
تحتوي على مجموعة من الأوامر الشكل (2). ولتشغيل أحد الأوامر نتبع إحدى
الطرق الثلاث التالية :

أ. استخدام مفاتيح الأسهم ↓ أو ↑ لتحديد الأمر ثم نضغط مفتاح
الإدخال Enter .

ب. الضغط على الحرف المميز من اسم الأمر .

ج . باستخدام الفارة بوضع مؤشرها على الأمر المطلوب والنقر على الزر
الايسر .



الشكل (2) أوامر احد القوائم

وقبل ان نتعرف على وظيفة كل امر من أوامر TP 7 لا بد من الاشارة إلى انه هناك مواصفات عامة لاسلوب العمل في بيئة الباسكال :

(1) وجود ثلاثة نقاط امام الامر : هذا يعني ان الامر يحتاج إلى تفاصيل معمقة اكثر لتنفيذه وبالتالي عند اختيار الأمر يظهر على الشاشة صندوق يدعى صناديق الحوار (Dialog Boxe) نستطيع تحديد بعض الخيارات اللازمة ، وستتناول صناديق الحوار في فقرة مستقلة .

(2) وجود العلامة ► جانب الأمر ، هذا يعني ان هنالك قائمة فرعية تظهر عند تنفيذ هذا الأمر .

(3) هناك بعض الأوامر لوفا باهت أو يختلف لون أحرفها عن لون باقي الأوامر: هذا يعني إن الأمر متوقف عن العمل مؤقتا ريثما تتوفر بعض الشروط الملائمة ليأخذ لونه الطبيعي وبالتالي امكانية تنفيذه ، مثل الامر Cut قبل تحديد النص المختار لقصه .

4) وجود اسماء مفاتيح امام بعض الأوامر ، هذه المفاتيح هي عبارة عن اسلوب مختصر لتشغيل الامر من غير الرجوع إلى القائمة .

(11-4) صناديق الحوار Dialog Boxes

بشكل عام صندوق الحوار يأخذ شكل النافذة ويحتوي على علبة الاقفال [.] Close Bar التي تساعد على اقفالها (إلغائها) في حالة استخدام الفأرة . ولا تحتوي على شرائط التدرج وعلبة تغيير الحجم ، ويضم أي صندوق حوار بعض أو كل العناصر التالية :

أ) علب الاختيار [] Check Boxes

تعطي امكانية اختيار اكثر من علبة في نفس الوقت ، وعند اختيار العلبة التي نريدها يتم وضع اشارة X داخل القوسين [] .

ب) أزرار الخيارات () Radio Buttons

لا يمكن اختيار اكثر من زر ضمن الجدول الواحد ، وعند اختيار احد الازرار التي نريدها يتم وضع نقطة في وسط القوسين () .

ج) أزرار الضغط Push Buttons

تستخدم لابلغ TP 7 ماذا سيفعل بالخيارات والقرارات التي حددناها من صندوق الحوار وعادة توجه ثلاثة ازرار :

OK لتشغيل الخيارات والقرارات المتخذة

Cansel إلغاء العملية كلها .

Help طلب مساعدة وتوضيحات حول علبة المحادثة .

د) جداول الاختيار ↓ List Options

عندما نريد إدخال بيانات ونحتاج إلى مساحة اكبر من المساحة المخصصة في علبة الحوار ، أو أننا نريد اختيار اشياء مخفية من المساحة المخصصة في صندوق الحوار، نلاحظ وجود السهم ↓ جانب المساحة المحدودة وظيفته

إظهار المخفي من المساحة المخصصة أو توسيع المساحة لامكانية الادخال .

كيفية التنقل داخل صندوق الحوار :

1. نستخدم مفتاح TAB أو Shift + TAB للانتقال من زر ضغط إلى آخر .

2. نستخدم مفاتيح الاسهم للانتقال من علة اختيار إلى أخرى ومن زر خيار إلى آخر .

3. نستخدم المفتاح TAB للانتقال من جدول أو قسم إلى آخر في صندوق الحوار.

(5-11) بعض المهمات التي يمكن القيام بها :

(1-5-11) فتح ملف جديد :

نختار الامر New من قائمة File ، عندئذ تظهر نافذة تنقيح جديدة بالاسم الافتراضي : NONAME00.PAS ، وتظهر المشيرة Cursor في الزاوية العليا اليسرى ، يمكننا البدء بكتابة تعليمات وأوامر البرنامج المراد ادخالها عند المشيرة .

(2-5-11) فتح ملف مخزن مسبقا :

نختار الامر Open من قائمة File ، عندئذ يظهر صندوق حوار باسم Open a File انظر الشكل (3) والذي يتألف من :

1 - حقل Name لإدخال اسم الملف المراد فتحه .

2 - حقل Files يضم جميع الملفات ، والفهارس المتوفرة على الديسك

الصلب Hard Disk ، يمكننا بالنقر المزدوج على

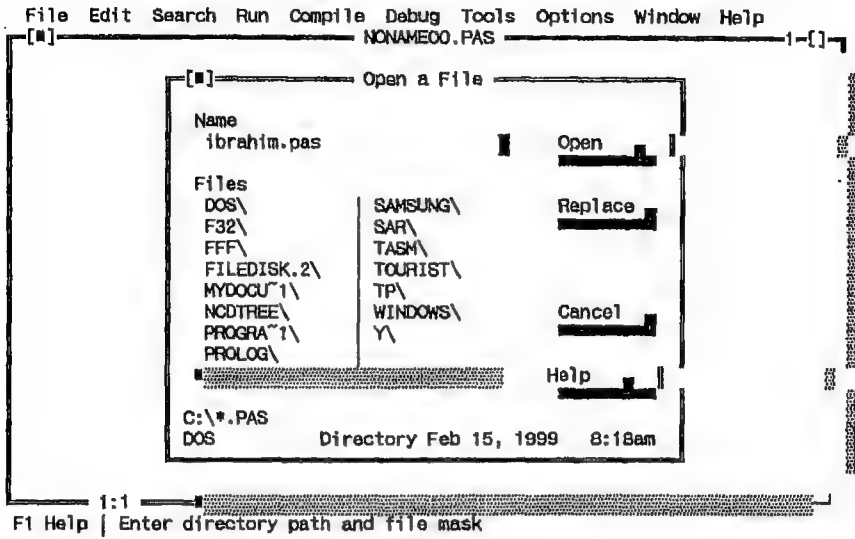
الفهرس الذي يحوي ملفات باسكال عندئذ تظهر جميع

الملفات ، وبالنقر على احد الملفات المراد فتحها ينتقل

الاسم إلى حقل Name ثم نستخدم الزر Open لفتح

(أو يمكن بالنقر المزدوج على اسم الملف) .

- 3 - الأزرار Open لفتح الملف المختار .
 Replace لاستبدال الملف المختار بملف آخر .
 Cancel إلغاء صندوق الحوار Open وجميع التعديلات التي تمت داخله
 Help إعطاء بعض المعلومات عن كيفية التعامل مع صندوق الحوار
- 4 - حقل في أسفل الصندوق يظهر بعض المعلومات عن الملف المختار .



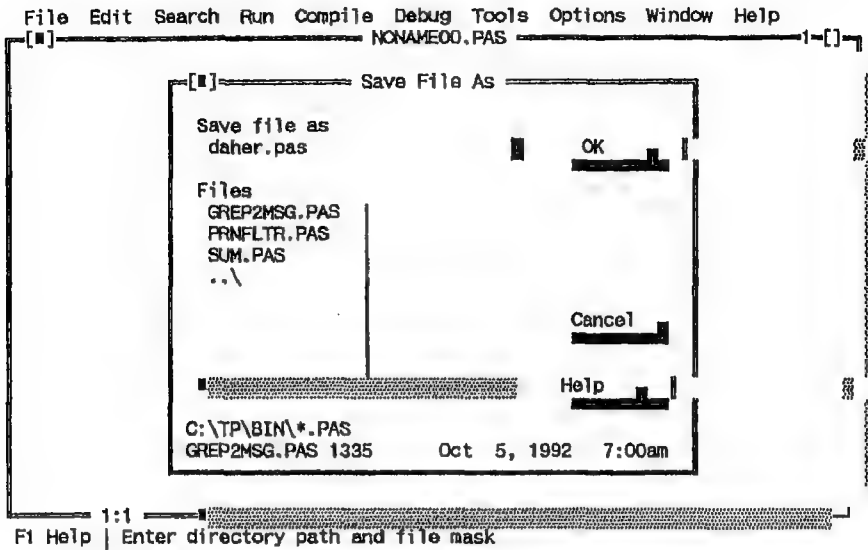
الشكل (3) صندوق حوار فتح (Open File)

- (11-5-3) تخزين ملف باسم جديد
- نختار الامر Save As من قائمة File ، عندئذ يظهر صندوق حوار باسم Save As ، انظر الشكل (4) ، والذي يحوي على :
1. حقل Save File As : لإدخال الاسم الجديد للملف .
 2. حقل Files : يضم الفهارس المتوفرة ، فإذا أردنا تخزين الملف على فهرس معين أو في سواقة معينة فعندئذ نقر اسم الفهارس أو السواقة.
 3. الأزرار OK : لإتمام عملية التخزين .

Cancel لإلغاء صندوق الحوار حقل Save File As وجميع التعديلات التي تمت داخله .

Help إعطاء بعض المعلومات عن الملف المراد تخزينه .

4. حقل في الأسفل يظهر بعض المعلومات عن الملف المراد تخزينه .



الشكل (4) صندوق حوار تخزين ملف Save As

(4-5-11) تخزين ملف بإسم قديم

نختار الأمر Save من قائمة File ، عندئذ يقوم الحاسوب بتخزين الملف مع جميع التعديلات التي تمت عليه تحت الأسم القلمم الموجود في شريط العنوان .

(5-5-11) طباعة ملف

نختار الأمر Print من القائمة File ، فيتم طباعة الملف المفتوح الموجود داخل النافذة الفعالة ، أما اذا أردنا طباعة مقطع معين من النافذة ، عندئذ نضغط المفاتيح K + Ctrl ثم المفتاح P .

(11-5-6) استخدام ميزة النسخ والقص واللصق والمسح عند كتابة البرنامج

لاستخدام هذه الميزات يجب اتباع ما يلي :

1- تحديد المقطع المراد تطبيق عليه ميزة النسخ أو القص أو المسح ويتم ذلك
بأحدى الطريقتين :

أ) باستخدام الفأرة : نضع مؤشر الفأرة في بداية المقطع المراد تحديده ثم
نضغط الزر الأيسر مع السحب على المقطع المراد تحديده .

ب) باستخدام لوحة المفاتيح : نضع المشيرة (إشارة الأدراج) على بداية
المقطع ثم نضغط مفتاح shift مع أحد مفاتيح الأسهم بالاتجاه
المناسب .

2- بعد تحديد المقطع يمكننا من استخدام أحد الأوامر التالية من القائمة Edit

Cut حذف المقطع المحدد ووضعه في الحافظة clipboard .

Copy وضع نسخة من المقطع المحدد في الحافظة clipboard .

Clear حذف المقطع المحدد نهائيا ، دون وضعه في الحافظة clipboard

3- لاستخدام المقطع المنقول إلى الحافظة بواسطة الأمر cut أو copy ، نقوم
بوضع المشيرة (مؤشر الأدراج) في المكان المراد وضع المقطع ثم نختار الأمر
paste فيتم نسخ المقطع الموجود في الحافظة .

بالإضافة الى ذلك هناك بعض المفاتيح في لوحة المفاتيح Keyboard تساعد
المستخدم على العمل داخل نافذة باسكال ، وهذه المفاتيح سنعرضها في الجدول
التالي :

اسم المفتاح	وظيفته
	الانتقال حرف واحد في كل مرة إلى اليمين
	الانتقال حرف واحد في كل مرة إلى اليسار
Ctrl +	الانتقال كلمة واحدة في كل مرة إلى اليمين
Ctrl +	الانتقال كلمة واحدة في كل مرة إلى اليسار
HOME	نقل المؤشر إلى بداية السطر الحالي
END	نقل المؤشر إلى نهاية السطر الحالي
Ctrl+Home	نقل المؤشر إلى أول سطر في النافذة
Ctrl+END	نقل المؤشر إلى آخر سطر في النافذة
Pg Up	نقل المؤشر بمقدار النافذة المفتوحة إلى الأعلى
Pg Dn	نقل المؤشر إلى بمقدار النافذة المفتوحة إلى الأدنى
Ctrl+Pg Dn	نقل المؤشر إلى بداية الملف
Ctrl+Pg Dn	نقل المؤشر إلى نهاية الملف
Del	حذف الحرف الواقع فوق المؤشر ويستخدم لحذف سطر فارغ بعد نقل المؤشر إلى بداية السطر الفارغ .
Backspace	حذف الحرف الواقع على يسار المؤشر
shift+الازرار	طباعة الحرف الكبير
Tab	نقل المؤشر مسافة على الأغلب بمقدار 4 أحرف

(11-5-7) تصحيح الملف قواعدياً وترجمته إلى لغة الآلة:

نستخدم الأمر **compile** من القائمة ، عندئذ يقوم الحاسوب بتدقيق صحة كتابة أوامر وتعليمات باسكال ، فإذا وجد خطأ في مكان ما في البرنامج عندئذ يعرض رسالة في أعلى النافذة برقم الخطأ ونوعه ، ويضع المؤشر على السطر الموجود فيه الخطأ .

(11-5-8) تشغيل البرنامج

نختار الأمر RUN من القائمة RUN ، عندئذ يبدأ الحاسوب بتشغيل البرنامج المفتوح على نافذة باسكال ويعطي النتائج على شاشة مستقلة تدعى شاشة النتائج .

(11-5-9) عرض نتائج تشغيل البرنامج

نختار الأمر user screen من قائمة Debug (وفي بعض إصدارات الباسكال يكون الأمر موجود في قائمة window) .

(11-5-10) الخروج من بيئة باسكال :

نستخدم الأمر Exit من القائمة file .

إذا كان الملف أو الملفات المفتوحة غير مخزنة أو أجري تعديلات عليها دون تخزين عندئذ تظهر رسالة :

TP\FF.PAS has been modified .		
Save ?		
<input type="button" value="Yes"/>	<input type="button" value="No"/>	<input type="button" value="Cancel"/>

هذه الرسالة تشير إلى أن الملف TP\FF.PAS يحتاج إلى تخزين ، فإذا أردنا تخزينه وتخزين التعديلات فنختار الأمر Yes وإذا كنا لا نريد تخزينه وتخزين التعديلات نضغط زر No ، وإذا رغبتنا بعدم الخروج من الباسكال نضغط cancel

المراجع العربية

1. ريان (أحمد)-خدمات الانترنت-منشورات المجمع الثقافي-أبو ظبي 1997
2. دوفور(ارنورد) انترنت-ترجمة-الدار العربية للعلوم-1998 .
3. نشاوي (محمد أسعد) تعلم ويندوز 95 باعتماد اللغة العربية-دار شعاع للنشر والعلوم-حلب 1997.
4. رحو (غازي ابراهيم)-الصباغ (عماد)-علم الحاسوب والبرمجة لطلبة الجامعات ، دار زهران للنشر والتوزيع 1997 .
5. برهان (محمد نور) رحو (غازي ابراهيم) نظم المعلومات المحوسبة-دار المناهج-1998 .
6. العبدالله (سعد الدين) الخوارزميات والبرمجة-منشورات جامعة حلب-1997.
7. الصفدي (محمد سالم) نظم معالجة البيانات والبرمجة بلغة باسكال-دار وائل للنشر-1999.
8. الشرايعه (احمد عبد العزيز) وآخرون الحاسوب والبرمجيات الجاهزة-دار وائل للنشر-1998.
9. نائب (ابراهيم عبدالواحد) رحو (غازي ابراهيم) - المرجع لمختبرات علم الحاسوب-دار المناهج 1998.
10. مجلة الكمبيوتر والاتصالات-سلسلة الاعداد لعام 97-98.
11. مجلة PC MAGAZINE ، سلسلة الاعداد للعام 97-98-99 .
12. دليل المستخدم Micro Soft Windows 3.1 شركة ميكروسوفت-1994 .
13. دليل المستخدم Micro Soft Excel 5 شركة ميكروسوفت-1993.

14. الفيومي (محمد احمد) مقدمة في علم الحاسوب والبرمجة بلغة باسكال - دار
الفلاح 1997 .
15. أكرم (كلير) - البرمجة بلغة الباسكال - دار ماكروهيل ، السدار الدولية
للنشر والتوزيع 1988 .
16. منصور (عوض) - أبو النور (محمد) - مقدمة في علم الحاسب الالكتروني
وبرمجة بيسك الأردن 1987 .
17. الراغب (فوزي) نظرة إلى أعماق الكمبيوتر منشورات المركز
الحديث للكمبيوتر 1985 .
18. الرماحي (سامي) الحاسوب (شرح تعليمي مبسط) الدار العربية
للعلوم 1988 .
19. طلبة (محمد فهمي) الحاسبات الالكترونية (حاضرها ومستقبلها)
موسوعة دلتا كمبيوتر 1992 .
20. رحو (غازي ابراهيم) الفيومي (محمد أحمد) الحاسوب وزارة التربية
العراقية العراق 1994 .

المراجع الاجنبية

1. Addyman , A. M. Et A (1979) , Adraft Description Of Pascal , Pascal New 14 And Software , Practice And Experience .
2. Atkinson , L. V. (1980)-Pascal Programming John Wiley And Sons Limited .
3. Atkinson , L. V. (1982)-A Students Guide To Programming In Pascal John Wiley And Sons Limited .
4. Barron D. W. (1981)-Pascal : The Language And Its Implementation John Wiley And Sons Limited .
5. Pindly W. And Watt Limited .
Pascal An Introduction To Methodical Programming Pitman Publishing Ltd.
6. Jenson, I And Wirths, N. (1975)
Pascal-User Manual And Report (Springier-Verlag, New Yourk) .
7. Matnszek , D. L. - Quik Pascal .
John Wiley And Sons Limited .
8. Moore (1980)- Foundation Of Programming With Pascal .
John Wiley And Sons Limited .
9. Pemerten S. And Danieis M. C.- Pascal Implementation .(1983)
10. Jones , R. M. (1983)
An Implementation Of Data Structure Allyn And Bacon Inc.
11. Schn Eider G. M. And Bruell Sec (1981)
Advanced Programming And Problem Solving With Pascal.
12. Weish , J , Sneeringer , W. J. And Hoare , C., A.(1977)

- Ambiguities And Insecurities In Pascal, Software Practice And Experience .
13. Wirth, N. (1975a)
An Assessment Of The Programming Language
Pascal Sigplan Notices,
 14. Wirth, N. (1975b) - Algorithms + Data Structure =
Programs
(Prentice-Hall Englewood Cliffe , N. J.)
 15. Wiison I. R. And Addsyman A. M. (1978)
A Practical Introduction To Pascal .
The Macmillan Presc LTD.
 16. Michael Shave - Data Structure
McGrow Hill (1975)
 17. Bertziss, A. T. (1975) - Data Structure Theory And
Practices .
Academic Press, London.
 18. Elson , M. (1973) - Concepts Of Programming
Languages
Sci. Research Ass, London.
 19. Knuth, D. G. (1973)
The Art Of Computer Programming VOL1, VOL2,
VOL3 Addisam – Wealeg London .
 20. Meek, B. And Heath , D. (Ed) (1980)
Guide To Good Programming Practice .
Ellis Horwood.
 21. John Beidler (1982) - An Introduction to Data
Structures
Allyn And Bacon
 22. Dijkstra , E. W. (1976) - A Principle Of Programming
Prentice – Hall
 23. Turbo Pascal – Elliot B. Koffman With Bruce Maxim
4th Edition – Addison – Wesley Publishing Company
(1996)
 24. Weizenbaum, Joseph, Computer Power And Human
Reason W. H. Freeman, San Francisco (1976)

25. Kernighan Brian W. And Dennice M. Ritchie The Programming Language Prentice-Hall, Englewood Cliffs NJ, (1978) .
26. Postel J. B. , Editor : Internet Official Protocol Standards RFC-1880 , STD-1, Internet Architecture Board,1995
27. Mitzi Walts , The Internet International Directory Zift Davis Press-1995
28. Colman P. And Dyson P. - Mastering Internet, The Windows 95/NT Edition , SYBEX 1997
29. Metzlar F.Scharp F.Scharp F.- Projects For Dos And Windows 3.1 , Benjamin / Comings – 1995 .
30. Larry Long – Nancy Long – Introduction To Computers And Information Systems – 5th ,Ed . , Printice Hall , Inc . UK ,1997 .
31. Slothink D.L.And Others – Computers And Applications- Second Edition , D.C. Meath And Company 1994 .
32. Marija Jnorusis – SPSS For Windows (Base System User's Gulde , Release 6.0)- Printed In The United Ststes Of America 1993 .
33. Crumlish Christian – The ABCS Of The Internet – SYBXX Inc., 1996 .

الإختبارات

الهدف :

تقديم بعض التمارين غير المحلولة وعلى الطالب حلها داخل
قاعة المحاضرة كاختبار لإظهار مقدرة الطالب في بعض مواضيع
هذا الكتاب .

الاختبار (1)

- 1- حول العدد التالي : $(57.125)_{10}$ إلى مكافئه في النظام الثنائي
والثماني والسادس عشري ؟

- 2- حول الأعداد التالية إلى النظام الثنائي :

$$(37.6)_8 =$$

$$(59ED)_{16} =$$

$$(462.7)_{10} =$$

الاختبار (2)

1- أوجد ناتج العمليات التالية :

$$(101101)_2 + (11101.11)_2 =$$

$$(573.42)_8 + (22.41)_8 =$$

$$(ED4.F)_{16} + (AA)_{16} =$$

$$(11001)_2 - (111)_2 =$$

$$(765)_8 - (32)_8 =$$

$$(B32)_{16} - (32)_{16} =$$

الاختبار (3)

1- أوجد المتمم الحسابي الثنائي الأول والمتمم الحسابي الثنائي الثاني لكل من الأعداد :

1- $(-53)_{10}$

2- $(36)_{10}$

باستخدام حاسوب طول الكلمة فيه يساوي 8 ثنائيات ؟

3- أوجد ناتج مايلي باستخدام حاسوب طول كلمته يساوي 8 ثنائيات ؟

$$(82)_{10} - (60)_{10} =$$

الاختبار (4)

- 1- أكتب خوارزمية وخريطة سير العمليات للمشكلة التالية :
يأخذ بائع عمولة على مجمل مبيعاته حسب ما يلي :
0.02 اذا كانت مجمل مبيعاته أقل من 500 دولار
0.05 اذا كانت مجمل مبيعاته من 500 دولار إلى 1500 دولار
0.15 اذا كانت مجمل مبيعاته أكثر من 1500 دولار
والمطلوب إيجاد مقدار العمولة بالعملة المحلية التي يأخذها البائع .

الاختبار (5)

- 1- أكتب خوارزمية وخريطة سير العمليات لإيجاد مضروب العدد N حسب القاعدة التالية :

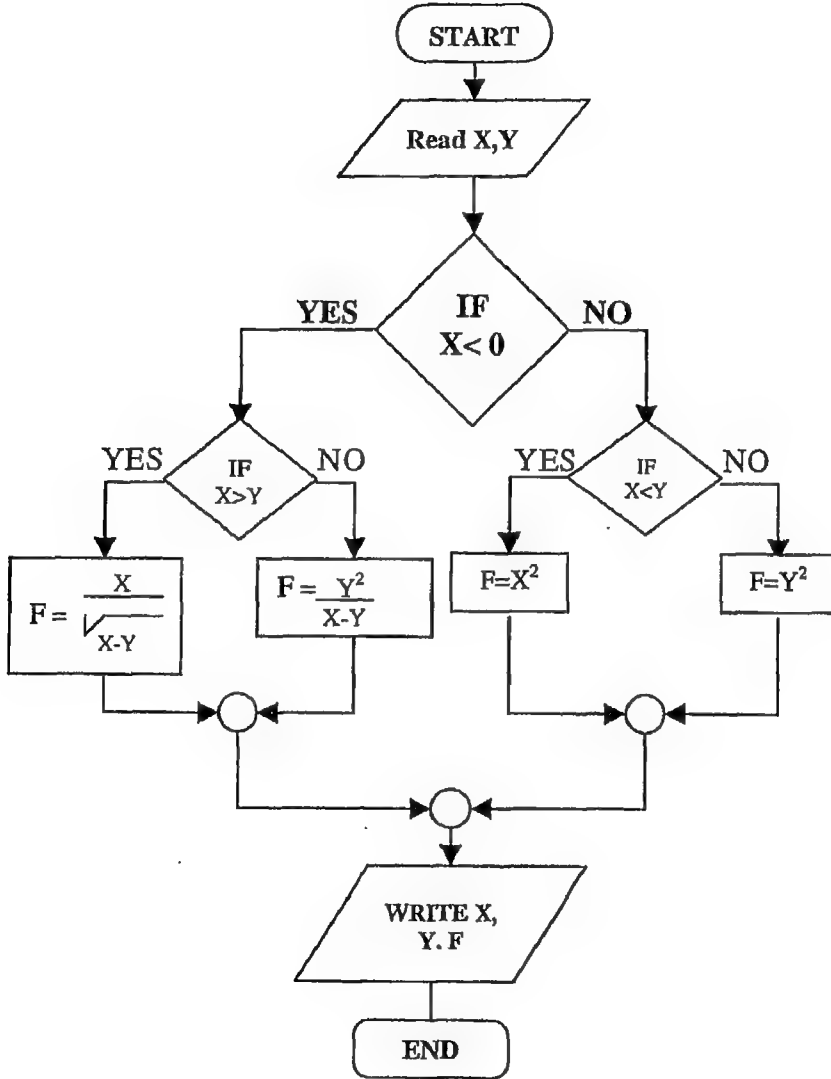
$$N! = \begin{cases} 1 & ; N = 0 \\ 1*2*3*.....* N & ; N \geq 0 \end{cases}$$

الاختبار (6)

أكتب برنامج بلغة باسكال لإيجاد عدد الطلاب الراسبين وعدد الطلاب الناجحين وعدد الطلاب الذين تزيد علاماتهم عن 70، لطلبة شعبة المقدمة حيث أن عدد الطلبة المتقدمين للامتحان يساوي 60 طالب .

الاختبار (7)

- 1- المخطط التالي يمثل مخطط سير العمليات لمشكلة ما ، والمطلوب كتابة برنامج بلغة باسكال المقابل لهذا المخطط ، وبين ما هي النتائج على افتراض أن البيانات المدخلة $x = 5$, $y = 3$.



الاختبار (8)

- 1- هل كتابة جمل وأوامر هذا البرنامج صحيحة ، بين الأخطاء الموجودة وصححها ؟

```
PROGRAM EXAM (INPUT,OUTPUT);
VAR
    X,Y : REAL
BEGIN
    X = 1.0;
    WRITELN ('X      Y');
    WRITELN ('---    --');
    WHILE X <= 6.0 DO
    BEGIN
        Y:=X*X*X ;
        WRITELN (X:5:2,' ',5:2);
        X:=X+0.5;
    END;
END.
```


الاختبار (9)

1- بين نتائج تنفيذ كل من البرامج التالية :

```
1) PROGRAM EX1 (Input,Output);
VAR
    M : REAL ;
    J,K,L : INTEGER ;
BEGIN
    J:= 4;
    K := J+2 ;
    L := J - 5 ;
    M := J- 6 ;
END.
```

النتائج

```
2) PROGRAM E2 (Input,Output);
VAR
    I,J :INTEGER;
BEGIN
    FOR I:= 1 TO 5 DO
        BEGIN
            I:= I + 1;
            I:= I * I;
            Writeln ('I=',I:4,'J=',J:4);
        END;
        Writeln ('*****');
    END.
```

النتائج :

الاختبار (10)

1- بين نتائج تنفيذ البرنامج التالي :

```
1) PROGRAM EX1(Input,Output);
   VAR
      X: ARRAY [1..5] OF REAL;
      I: INTEGER; SUM,A:REAL;
   BEGEN
      SUM := 0.0;
      FOR I:= 1 TO 4 DO
      BEGEN
         SUM :=SUM + X[I];
         Writeln (SUM,' ');
      END;
      A:= SUM/4;
      Writeln ('SUM =' ,SUM,'A=',A);
   END.
```

النتائج :

الاختيار (11)

أكتب برنامج بلغة باسكال لضرب مصفوفة A أبعادها (5×4) بثابت K ؟

الاختبار (12)

البرنامج التالي بداية برنامج لإيجاد أكبر عنصر في مصفوفة A أبعادها (4,4) ،

أكمل البرنامج بحيث يطبع أكبر عناصر المصفوفة المدخلة ؟

```
PROGRAM EXAMPLE (INPUT,OUTPUT);
VAR
  A: ARRAY [ 1..4] OF REAL ;
  I,J,MAX: INTEGER;
BEGIN
  WTITELN ('Enter Data of Matrix A :');
  FOR I:= 1 TO 4 DO
    BEGIN
      FOR J:= 1 TO 4 DO
        READ ( A[I,J] );
        WRITELN ;
      END;
```